NEAR WELL BORE STREAMLINE SIMULATION

MARJAN HASHEM

Near Well Bore Streamline Simulation

by

Marjan Hashem B.Sc

A thesis submitted to School of Graduate Studies in partial fulfillment of the requirements for the degree of

Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland

February 2011

St.John's Newfoundland Canada

Abstract

This thesis presents a methodology for generating and analyzing streamlines in reservoirs near wellbores using a finite difference method and the Pollock method. The proposed methodology has been applied in 2D using Cartesian and Polar coordinate systems.

This proposed method demonstrates how all potential geological and mechanical factors affect streamline behaviour and analyze their effect on route of stream path lines movement. By using this method, stream path lines can be visualized for any given boundary conditions. Subject to the basic assumptions of incompressible fluid and avoidance of gravity effects, there is an opportunity to simplify numerical equations for streamline simulation and retrieve more effective results. Fluids are moved along the natural streamline grids.

Permeability, well conditions and other process factors such as pressure, velocity and time of travel dictate the route and direction of streamlines. The advantages of this method include flow visualization, more efficiency and computational speed, and the ability to add of additional boundary conditions in various numbers of grid blocks. In particular, this method offers streamline visualization in inter-well connectivity.

Streamline simulation is increasingly employed by geoscientists and engineers in the oil and gas industry to model the flow of fluids in reservoirs. Flow simulation can run on fine-grid, high-resolution models in reasonable times on off the shelf hardware. In addition, streamline simulation identifies fluid flow paths between sinks and sources. This visual and quantifiable information allows geoscientists to examine model connectivity, as well as drainage and irrigation zones associated with producers and injectors. Also, it helps reservoir engineers analyze streamline migrations and predict targets for efficient drilling.

The Pollock's method generally is used for an orthogonal grid cell, however in reality most of the reservoirs near wellbore do not adjust in Cartesian modeling shape. Thus, the next advancement in this research explored ways to find stream path lines according to the Pollock method but with assuming Polar coordinates instead of Cartesian coordinates, when wells' location is in the center of the rings and all streamlines come from different directions and collect in the center of the circle.

My contribution in this study includes the MATLAB implementation of the following steps:

- Developing specific finite difference grid blocks according to given reservoir parameters
- Implementing numerical pressure solvers in both Cartesian and Polar coordinates

- Finding time of flight for streamline movement in grid blocks according to published methods

- Visualizing streamlines in 2D by finding streamline entrance and exit points in each grid cell and connecting all points by line segments.

Most of the mathematical formulations are taken from available literature and other documents. This methodology is demonstrated through the use of three case studies for both Cartesian and polar coordinates. The research includes two main topics: Streamline simulation in Cartesian and Polar coordinates. Each topic has two main parts:

- Theoretical part; in this part, all numerical and mathematical modeling is described, using finite difference methods for pressure solving.

- MATLAB code programming; in this part, a MATLAB program is written according to theoretical equations.

The MATLAB code is included in an appendix. This MATLAB code can visualize all streamlines with different boundary conditions, various process and geological factors in different reservoir areas. The graphs which are resulted from this numerical method can provide valuable information for engineers to have better understanding of fluid flow in the reservoir.

Regarding research novelty, streamline simulation in near well bore regions using radial geometry has not been done before. Numerical methods are effective tools to optimize time and costs in oil and gas projects and reduce uncertainties.

Finally this research has significant potential applications in the future to improve fluid flow modeling near well bores. The numerical method discussed in this study can be extended to three dimensional coordinates. In this study fluid has been assumed to be incompressible. The method can be used with compressible fluids in future, which is closer to reality.

Acknowledgement

I would like to express my deepest gratitude and appreciation to my supervisor Dr. Thormod Johansen, whose encouragement, supervision and guidance sparked my interest and enabled me to achieve extraordinary experiences throughout every stage of this research. He continually and convincingly conveyed a spirit of adventure and excitement with respect to research, scholarship, and teaching. Without his guidance and persistent help this dissertation would not have been possible.

I also would like to thank the Faculty of Engineering and Applied Science of Memorial University of Newfoundland and the School of Graduate Studies for giving me the opportunity of joining the Memorial University family and having extraordinary experiences during these years.

Words fail to express my appreciation to my lovely husband, Nasser Daiyan, whose dedication, honest love and persistent confidence in me has taken the load off my shoulders. I owe him for unselfishly letting his intelligence, passions, and ambitions collide with mine. He is the best love and friend that I have.

Where would I be without my family? My deepest gratitude goes to my family for their unflagging love and support throughout my life; this dissertation is simply impossible without them. My Father, Ali Akbar Hashem, the best father in the world, is forever remembered. I feel his presence with me every moment of every day; I am sure he shares our joy and happiness in heaven. I cannot ask for more from my mother, Sedigheh Mansour, as she is simply perfect. I have no suitable words that can fully describe her everlasting love. All I have is because of her love, dedication and prayers. I am also grateful to my brother, Majid, and sisters, Shirin & Shohreh, for their tremendous support and encouragement along the way.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of my program.

Table of Content

Page

Chapter I

Introduction

Chapter II

Literature Review

Table of Content (Cont.) **Page**

Chapter III

The Pollock Method for Streamline Generation

Chapter IV

Pressure Solution in Cartesian Geometries

Chapter V

Pressure Solution in Radial Geometries

Table of Content (Cont.) **Page**

Table of Content (Cont.) **Page**

Chapter VI
Case Studies

Chapter VII
Conclusions

List of Figures and Illustrations

Figure **Page**

Figure (Cont.) **Page**

List of Symbols, Abbreviation and Nomenclature

Abbreviations:

EOS equation of state

RC reservoir condition

Greek Symbols

α Liquid volume fraction or liquid holdup

β Water volume fraction

μ Viscosity

$\bar{\rho}$ Average two phase density

ρ Density

Symbols

A Cross- sectional area

D Diameter

D_h Hydraulic diameter

f Friction factor

g Gravitational acceleration

K Absolute permeability

L Length

m Mass flux

N Number of finite difference grid cell segments

p Pressure

Q Flow rate

q Volumetric Flux

r Radius

r_w Well bore radius

r_e Well outer radius

t Time

u Darcy's velocity or volumetric flux

V Volume

v Velocity

Subscript

I In-flow

i, j, k Index for phase or components

o Oil phase

p Particle

res Reservoir or reservoir condition

x x- Direction

y y- Direction

z z- Direction

Chapter I

Introduction

Background

Streamline simulation of displacement processes in oil reservoirs gained industrial attention during the 1980s and 1990s, when the method was implemented as a 3D procedure (Bratvedt et al. (1993)). Originating from the more complicated and less applicable 3D front tracing method (Bratvedt et al. (1993) and Glimm et al. (1983)), it offered the opportunity of incorporating operator splitting techniques to include the effects of fluid compressibility, diffusion and gravity effects.

As the front tracking methods were for purely hyperbolic systems only, such operator splitting techniques used a three step strategy to solve parabolic problems:

First, the 3D pressure equation is solved and streamlines generated according to the pressure field.

Secondly, the fluids are moved along the computed 1D streamlines over a time step ignoring gravity.

Thirdly, a gravity correction step is made where the fluids of different densities are segregating vertically.

The success of this streamline approach lies in the fact that operating splitting is mathematically rigorous, together with the fact that fluid transport along streamlines does not need to be based on the assumptions of zero diffusion and incompressibility.

Furthermore, streamlines can be updated at any time, for example if new wells are drilled. In fact, streamlines may be updated on each time step in which case a streamline method reduces to an adaptive grid finite difference method., When streamlines are updated on each time step the computational advantage, in terms of CPU timesavings, is lost.

In spite of the successful application of operator splitting techniques, streamline simulation has limitations and can never replace conventional simulation techniques. First of all, this is because the splitting of a pressure equation from the viscous/ diffusive part of the problem is not accurate if the fluids are highly compressible and/or when gravity effects dominate over viscous transport mechanisms. Therefore, streamline simulation is mostly used for water flooding in oil reservoirs and less for gas injection and recovery of heavy oil.

Another limitation of the streamline approach is that the numerical method used for streamline generation (the Pollock method) tends to be numerically unstable near well bores. Therefore, detailed streamline modeling of near-well flow is absent from the published literature. The work presented here focuses on this aspect of streamline modeling.

1-1 Problem Identification

The traditional Pollock method for streamline generation is for Cartesian geometries. It is a numerical procedure, which takes a streamline entrance point in a grid block as input and calculates the exit point based on the pressure in the grid block itself and the pressures in the adjacent grid blocks. This information is available from a priori solution of the pressure equation (Laplacian). Figure 1.1 depicts the procedure.

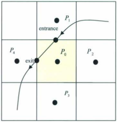


Figure 1.1: Exit Point (Calculated)

Once the exit point is calculated, a straight line-draw is made to connect the exit and entrance point to constitute a segment of a large-scale streamline.

The Pollock method is described in detail in Chapter III. Here, it suffices to identify the lack of accuracy inherent in the method as applied to near well situations. Referring to Figures 1.1 and Figure 1.2, the fluid velocity across the faces of the grid block can be approximated:

The velocity, V_1, across the upper face is calculated from Darcy's Law using upscaled permeabilities and the pressures P_1 and P_0 in Figure 1.1. Similarly, the velocities V_2, V_3 and V_4 across the other faces are calculated (Figure 1.2).

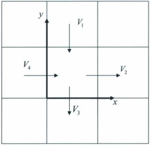


Figure 1.2: Velocity Across the Four Faces of Grid Cell

Each of the velocities V_i are assumed to be constant over the face i, however, the horizontal x and vertical y velocities are assumed to vary linearly inside the grid block, i.e. it is assumed that:

$$V_H(x) = ax + b \qquad (1\text{-}1)$$
$$V_V(y) = cy + d$$

where the constants a, b and c, d are chosen to match the known velocities V_1, V_3 and V_2, V_4 respectively.

An obvious limitation of this procedure is that it does not allow an entrance and an exit point to be located on the same face (Figure 1.3). Therefore, grid blocks must be small to capture this effect. In near-well bore situations, a radial geometry mitigates this disadvantage since grid blocks in a cylindrical grid are small near the well bore.

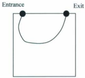


Figure 1.3: Entrance and Exit Point

Such situations can occur close to well bores, for example, when fluids have to converge into perforations in the presence of small but significant heterogeneities.

Another problem with the Pollock method is that for near-well flow, the flow velocity does not vary linearly as linear flow geometries as shown in equation (1-1). Therefore, consider a cylindrical grid, as shown in Figure 1.4. As the fluids approach the well bore, the velocity increases, however not linearly.

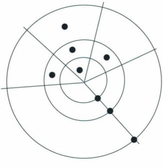


Figure 1.4: Cylindrical Grid Cells

This follows from the fact that the pressure decreases logarithmically towards the well bore (assuming purely radial):

(1-2) $$P(r) = a \ln r + b$$

According to Darcy's Law:

(1-3) $$V = Const \frac{\partial P}{\partial r} = Const \frac{a}{r}$$

When we also allow angular flow in Figure 1.4, we must allow fluids to "leak" into the neighboring blocks in the angular direction. Instead of equation (1-3), we therefore assume;

$$V(r) = \frac{a}{r} + b \tag{1-4}$$

where the constants a and b are chosen to match the known radial velocities over the faces of the block in the r-direction.

In the angular direction, the assumption of linear change is adequate:

$$V(\theta) = c\theta + d \tag{1-5}$$

where again the constant c and d are determined from the known face velocities. The primary objective of this work is therefore to implement the modified Pollock method in Polar coordinates to better adapt to the near-well geometry.

The streamline generation method discussed above is based on knowing the pressure distribution a priori. This is obtained by solving the Laplacian. In this work, we are primarily focusing on the method as applied to the near well region, i.e. we need to solve the Laplacian in a cylindrical geometry. In reservoirs, we can not impose an assumption of homogenous and isotropic medium.

It is a fact that the general (anisotropic) Laplacian in a cylindrical geometry frequently is misformulated in the petroleum literature. This is because a non-tensorial formulation is highly desirable for simplicity reasons. Therefore, the concept of angular direction permeability is often used. However, this entity does not exist in an anisotropic medium and consequently a full tensorial formulation cannot be avoided. A secondary objective of this work is therefore to rigorously derive the appropriate Laplacian for near well streamline modeling. This is presented in Chapter VI .

1-2 Well Productivity Modeling

In conventional reservoir models, the wells are represented as sources/sinks within individual grid blocks. The well segment penetrating a grid block is assumed to be aligned with the block boundaries (faces). This was an adequate approach before 1990 where all wells in fields were vertical (Figure 1.5).

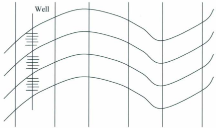


Figure 1.5: Wells in Fields

However, during the 90s, formidable technology developments were made within drilling and completion of advanced wells specifically lateral wells, and drilling wells of 2000-3000 m long production sections became common practice. The traditional well representation for such wells can be completely inadequate as shown in Figure 1.6.

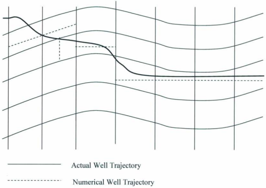


Figure 1.6: Numerical and Actual Well Trajectory

Clearly, representation of advanced wells in reservoir simulation therefore needs considerable improvements. By using streamline modeling near well bores, the details of the flow pattern can be captured and the well productivity can be accurately calculated by integrating flow along streamlines. These local productivity models can subsequently be incorporated into standard simulator models.

The work presented here is the first step to achieve the ultimate goal of this research: To develop accurate, local productivity models, which incorporate near well heterogeneities, anisotropy and well completion details for complex well trajectories.

1-3 Objectives of the Research

The objectives of this work are:

1-To rigorously derive and describe the equations for 2D streamline simulation in both Cartesian and cylindrical geometries

2- To implement streamline simulation in two dimensions in Cartesian coordinate and Polar coordinate systems.

3- Develop MATLAB programs for both Cartesian and Polar coordinates for streamlines simulation near well bores.

4- To investigate flow behavior using the streamline models in heterogeneous isotropic reservoirs with different reservoir conditions and well conditions.

5- To investigate effects of various boundary conditions on stream line behaviour.

6- To provide recommendations on further development of the models.

1-4 Thesis Organization

This research attempts to provide a methodology for evaluating streamline behaviour near well bores by finding stream path lines in the reservoir. A comprehensive model is developed based on locating the entry and exit points of each streamline in each given grid cell using the Pollock method for Cartesian coordinate system and a modified Pollock method for cylindrical geometries.

This thesis is divided into seven chapters. The first chapter provides an introduction to the thesis, background information, objectives of research, scope of study and thesis organization.

Chapter II presents literature review, previous background works and researches on streamline simulation. Then, variation of streamline modeling is reviewed.

In Chapter III, there is a comprehensive explanation of the Pollock method. Finite difference grid cells and the structure of the grid blocks are introduced. In addition, fluid flow modeling is introduced. In addition, formulation and equation series for computation of flow modeling are described.

Chapter IV presents the pressure solution in Cartesian coordinates. This chapter provides calculation process for finding the pressure distribution across the given reservoir by solving Laplacian equation.

Chapter V presents solution for the pressure distribution in radial coordinates and shows step-by-step calculations to develop a MATLAB code to simulate streamlines in reservoirs in radial coordinate system.

Chapter VI presents case studies in both Cartesian and Polar coordinate system. Three case studies in Cartesian coordinates and two case studies in Polar coordinates with various boundary conditions are presented.

The thesis is summarized and concluded in Chapter VII where research novelty and recommendations for future research are discussed.

Chapter II

Literature Review

The current popularity of streamline simulation should more aptly be termed resurgence, given that streamlines have been in the literature since the paper by Muskat et al. (1934). The method has received repeated attention since then, and over the last 70 years different ideas and applications have been published about streamline simulation as discussed in this chapter, e.g. Thiele (2001).

2-1 What is Streamline Modeling?

Streamlines, pathlines and streaklines are convenient tools for describing and visualizing flow given by an external flow velocity field $\overline{V}$:

(2-1) $$\overline{V} = [V_x, V_y, V_z]$$

Streamlines are a family of curves S that are instantaneously tangent to the velocity vector $\vec{V}$ at every point;

$$\frac{d\vec{s}}{dt} = \vec{V} \tag{2-2}$$

Streamlines can be traced for any vector field, although the most common is that $\vec{V}$ represents a velocity obtained from the solution of a set of flow equations. For incompressible flow, streamlines defined at a single instant do not intersect and cannot begin or end inside the medium except at singularities (sources and sinks). Streamtubes are regions bounded by streamlines. Because streamlines are tangent to the velocity field, a fluid that is inside a streamtube must remain within the same streamtube.

A pathline *x(t)* is the trajectory traced out by an imaginary massless particle following the flow of the fluid from a given starting point,

$$\begin{aligned} \frac{dx}{dt} &= \vec{V} \\ x(t_0) &= x_0 \end{aligned} \tag{2-3}$$

A streakline is the locus at a given instance of the positions of all fluid particles that have gone through a fixed spatial point in the past. In steady state flow streamlines, streaklines and pathlines coincide. In an unsteady flow they can be different. In this work we consider only steady state flow. Streamline simulation is a technique that predicts multi-phase displacements along the streamlines generated from numerical solutions to the Laplace equation. The technique decouples computation of saturation variation from the computation of pressure variation in time and space. Using a finite difference method, the initial steady state pressure field is computed based on spatial variations in mobility, and is updated in response to significant time-dependent changes in mobility. The flow velocity field is then computed from the pressure field, and streamlines are traced based on the underlying velocity field. Streamlines originate at the injectors (source) and culminate at producers (sink). Once the streamline paths are determined, displacement processes are computed along the streamlines using 1-D, analytical or numerical models (Thiele (1994), Batycky (1997)) and ResAssist (http://www.res-assist.com/).

The integration of equation (2-3) to obtain particle paths and/or travel times is known as particle tracking, for which there exists extensive literature. The particle tracking literature is primarily concerned with problems where the velocity field is only known at a finite set of points, either measured or calculated from a flow model, and interpolation is needed to integrate pathlines. In computational fluid dynamics, particle tracking has been used for visualization (Kipfer et al. (2003), Knight et al. (1996), Sadarjoen et al. (1997), Shirayama (1993) and Strid et al. (1989)).

Velocity interpolation in control-volume mixed finite-element methods is a related subject to particle tracking and has been considered in some papers such as Naff et al. (2002). Within groundwater flow simulation, particle tracking is used to model contaminant transport (Oliveira et al. (1998), Cordes et al. (1992), Schafer-Perini et al. (1991) and Shafer (1987)).

The integration of equation (2-3) for visualization, is usually done numerically using a Runge–Kutta type solver, whereas in groundwater flow, semi-analytical integration is the most common. In some research, streamline tracking is a subset of particle tracking, since streamlines may be computed by particle tracking if the streamline parameters are introduced as an artificial time variable for which the instantaneous velocity field $\overline{V}$ is steady.

Most researchers consider streamline tracing in the context of streamline simulation of flow in hydrocarbon reservoirs (Batycky (1997), Bratvedt et al. (1993) and King et al. (1998)). In some cases, the fluid velocity $\overline{V}$ is typically given as the numerical solution of a set of flow equations for $\overline{V}$ and the fluid pressure p of the form;

$$(2\text{-}4) \qquad \begin{aligned} c\frac{\partial p}{\partial t} + \nabla.\overline{V} &= b \\ \overline{V} &= -a(\overline{x})\nabla p \end{aligned}$$

The two equations are commonly referred to as the pressure equation and Darcy's law, respectively. The corresponding discrete velocity approximation depends on the numerical method:

- For finite-difference methods, the pressure is usually computed at cell centers, and fluxes can be obtained at cell edges by application of a discrete form of Darcy's law (Zheng et al. (2002)).
- For finite-element methods, the numerical solution gives a continuously defined pressure approximation as the sum of the basis functions for all elements weighted by the corresponding node values. Although a continuously defined velocity can be

obtained from Darcy's law, a better strategy where continuous fluxes are obtained at cell edges, is given in some literatures (Durlofsky (1994) and Kinzelbach et al. (1992)).

- Mixed finite-element methods solve for velocity and pressure simultaneously, resulting in a more accurate velocity field than finite difference and standard finite element. The continuously defined velocity is given by the degrees of freedom at the edges and the corresponding basis functions (Kaasschieter (1995) and Durlofsky (1994)).
- Finite-volume methods include multi-point flux approximations and control-volume finite-element methods (Verma et al. (1997), Aavatsmark (2002) and Edwards (2002)).

In these methods fluxes are computed at cell edges. In other words, a continuously defined velocity field is obtained only for the mixed finite-element method. For the other methods one must use an interpolation scheme to determine the velocity from the discrete fluxes at the cell edges.

In reservoir simulation and groundwater flow, the predominant way of computing streamlines is by use of a semi-analytical technique. In semi analytical methods, the interpolation of the velocity is simple enough that analytical integration is possible within each grid cell. As an example, let us consider the popular Pollock method, (Pollock (1988)) which is described in more detail in Chapter III. Given an entry point of a streamline into a grid cell, Pollock's method starts by mapping the grid cell onto the unit square (or unit cube in 3D). Each component of the velocity field is then approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit in each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point and the exit point is mapped back into physical space to give the entry point into the next cell, and so on. In this method, incompressible fluids are assumed and gravity effects are ignored. Thus, severe limitations are associated with this methodology. However, as will be discussed later, these assumptions can both be relaxed using operator splitting and streamlines can then be approximated.

Flow phenomena are governed by Partial Differential Equations (PDEs) involving functions of space and time. Most single-phase flow models in engineering are partial differential equations of second order and are often linear, while multi-phase flow models are highly non-linear.

A general second order linear partial differential equation in two Cartesian variables can be written as;

$$A(x,y)\frac{\partial^2 u}{\partial x^2} + B(x,y)\frac{\partial^2 u}{\partial x\,\partial y} + C(x,y)\frac{\partial^2 u}{\partial y^2} = f\left(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right) \tag{2-5}$$

Three main types arise, based on the value of the discriminant, D, $D = B^2 - 4AC$

Hyperbolic, wherever *(x, y)* is such that $D > 0$;

Parabolic, wherever (x, y) is such that $D = 0$;

Elliptic, wherever (x, y) is such that $D < 0$.

The complete solution of a PDE requires additional information, in the form of initial conditions (values of the dependent variable and its first partial derivatives at $t = 0$), boundary conditions (values of the dependent variable on the boundary of the domain) or some combination of these conditions.

Streamlines exist only for elliptic problems when the following conditions are satisfied:

- Single phase flow
- Incompressible fluid
- No gravity (Horizontal flow)
- Zero dispersion

However, reasonable approximations to other situations have been prepared which will be discussed below.

2-2 Variation of Streamline Modeling

2-2-1 Eulerian-Lagrangian Methods

For multi-phase flow, the governing PDEs are highly non-linear in the unknown pressure and saturations. Some researchers used Eulerian-Lagrangian methods in a sequential approach, first solving a pressure equation and subsequently a saturation equation for two-phase immiscible flow (Dahle et al. (1990), Dahle et al. (1992), Dawson (1991), Douglas et al. (1997) and Espedal et al. (1999)).

The Eulerian-Lagrangian methods are widely used to observe and analyze fluid flows, either by observing the trajectories of specific fluid parcels which yields what is commonly termed a Lagrangian representation, or by observing the fluid velocity at fixed positions which yields an Eulerian representation. Lagrangian methods are often the most efficient way to sample a fluid flow and the physical conservation laws are inherently Lagrangian since they apply to moving fluid volumes rather than to the fluid that happens to be present at some fixed point in space. Nevertheless, the Lagrangian equations of motion applied to a three dimensional continuum are quite difficult in most applications. Therefore, almost all theories in fluid mechanics are developed within the Eulerian system. Lagrangian and Eulerian concepts and methods are used side-by-side in many investigations.

The Eulerian-Lagrangian methods require a tracking algorithm to trace a streamline from the starting point until the exit boundaries. The Lagrangian method calculates streamlines locally around specific group points on a fixed grid. Thus to advect the solution along the local streamlines, front-tracking is used. In the Eulerian method the advected solution is projected back onto a fixed grid cell and then used as initial condition for the equation (Ask, et al. (2000)).

2-2-2 Particle Tracking Technique

A popular method for tracing streamline in ground water is the particle tracking technique. Particle tracking techniques are important methods for petroleum reservoir and ground water aquifers. Normally, the velocity for each grid block is found prior to generation of streamlines and the streamline computation is done by semi-analytical methods (Pollock (1988), Goode (1990), Lu (1994), Datta-Gupta et al. (1995), Russel et al. (2000)), or less efficiently by ODE-solvers such as Runge-Kutta methods (Ask et al. (2000)).

The paper by Pollock (1988), (described later in more detail), was a breakthrough for this method. Since the velocity field is obtained numerically, the method can rigorously handle arbitrary geometries, rate imbalance and areal heterogeneities, (Datta-Gupta et al. (1994)). In addition, streamline methods used on tracer tests can help to find the preferred flow paths and include reservoir heterogeneities data that generally does not appear in a conventional pressure transient test. In addition, the particle tracking technique is designed to trace particles of the fluid flow field where the flow velocity is calculcated at a limited number of locations. Given a velocity field, a particle is traced one element by one element until either a boundary is encountered or the available time is completely consumed (Cheng et al. (1998)).

2-2-3 Streamline VS. Streamtube Modeling

A region bounded by streamlines is called a streamtube. In a steady flow, because the streamlines are tangent to the flow velocity, the fluid that is inside a stream tube must remain forever within that same stream tube.

Imagine a set of streamlines starting at points that form a closed loop, (Figure 2.1). These streamlines form a tube with impermeable walls since the walls of the tube are made up of streamlines. By definition, in the abscense of dispersion and gravity there can be no flow normal to a streamline. This tube is called a streamtube.

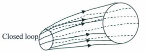


Figure 2.1: Streamtube- Schematic Figure

Streamline and streamtube technology, to a large extent, have been driven by the fact that heterogeneity controls recovery factors for most fields. This realization caused the derivation of more complex geological models. After Muskat et al. (1930) research about streamline modeling, streamtube technology was developed in the 1960s, (Higgins, et al. (1962), Leblanc, et al. (1971)). Two dimensional streamtube models were initially available for homogenous permeability regular flow patterns, such as five-spot patterns. Streamtube models were later generated for irregular well positions and areally heterogeneous reservoirs. Streamtube models only allow constant well rates and positions. Gravity effects, and therefore the vertical sweep efficiency are not accounted for. Thus, streamtube modeling often cannot allow for infill drilling or shut in of wells and streamlines are fixed in space.

From mass conservation, we see that for a steady flow, the mass-flow rate is constant along a streamtube. In a constant density flow, therefore, the cross-sectional area of the streamtube gives information on the local velocity. In the fast flow region, the streamlines come closer together, and the area between them decreases. Since the fluid density is constant, the velocity must increase according to the principle of mass conservation. For constant density flow, wherever the area between streamlines decreases, the velocity increases. This is similar to what happens with constant-density flow through a duct or a pipe; if the area decreases, the velocity increases so that the volume flow rate is maintained constant (volume flow rate out must equal volume flow rate in by continuity).

Before the current technology of streamline simulation developed in 1990s, several methods had been proposed. Muskat et al. (1930) considered streamline for 2D flow modeling. Fay et al. (1951) brought in "time of flight" for finding movement along streamlines in 2 dimensions. Higgins et al. (1962) used streamtube method and solved a 1D conservation equation in each streamtube.

The difference of streamline with streamtube simulation using the time of flight approach is that the flow can be determined by integrating along a streamline rather than first defining the streamtubes.

Using streamlines for solving the problem started from the 90s and became more popular in oil and gas research areas, (Datta-Gupta et al. (1995), Bratvedt et al. (1996), Hewett et al. (1997) and King et al. (1998)), and now streamlines based flow simulation is accepted as an effective and complementary technology to more traditional flow modeling approaches such as finite difference method (FD).

Batycky (1997) presented the development and application of a three-dimensional, two phases streamline simulator that is applied to field scale multi-well problems. A streamline simulator for the pressure field and for the saturation distribution consists of two factors:

1-stream path line tracking and

2-modeling streamline solution in 1D

A streamline model uses an IMPES approach i.e. implicit in pressure and explicit in saturation, which is computationally more effective than a fully implicit approach. As long as there is more than one phase, the equations are nonlinear. This is the main advantage of streamline models compared to conventional reservoir models.

2-3 Periodic Updating of Streamlines

Assuming a fixed steamtube/ fixed streamline assumtiption was one of the disadvantages of the streamlines method during 70's and 80's. Martin et al. (1979) and Renard (1990) considered changing streamlines and since the middle of 90's this assumption has not been used anymore (Thiele et al. (1996) and Batycky et al. (1997)). The main application of considering changing streamlines was for problems with changing well conditions and gravity. To define changing streamlines the problem was assumed steady state at each fixed time interval before being updated. This method worked well for mobility induced nonlinear case studies, but mapping analytical, self-similar hyperbolic solutions would not allow solving systems with changing well conditions and gravity (Theile et al. (1995), Theile et al. (1997), Theile (2001)). Another important factor required was the ability to solve transport problems with generalized initial conditions along each streamlines (Batycky et al. (1997).

Then streamline changing could be modeled while guaranteeing the fluid transportation in correct direction.

2-4 Numerical Modeling for Streamlines Simulation

Bommer et al. (1979) introduced numerical 1D solution for streamlines to solve a Uranium leaching problem. They used fixed streamline assumption for the numerical solution. Batycky et al. (1997), combined changing 3D streamlines with 1D numerical solution in time-of-flight (TOF)-space. This combination of the ideas had significant effects on using streamline-based simulation in reality where the streamlines change because of mobility differences and changing well conditions. Therefore, streamlines could adjust their initial conditions with the new location. i.e. the conditions existing at the end of the previous timestep. Using 1D numerical solution also made it possible to consider any 1D solution along streamlines, including complex compositional displacements or contaminant migration displacements, (Crane et al. (2000) and Thiele et al. (1997), Thiele (2001)).

2-5 Gravity

Gravity is another factor that needed to be considered in calculations. The velocity vector is the sum of the phase vectors; however, the phase vectors are not parallel when gravity exists. Bratvedt et al. (1996) presented a solution using the concept of operator splitting, an idea that had been used before in front tracking, (Glimm et al (1983), Bratvedt et al (1992)). This method solves the material balance equations in two steps:

A "convective step" along streamlines and a "gravity step" along gravity lines. In the second step, fluids are segregated in the vertical direction with regards to their phase densities.

2-6 Compressible Flow

The assumption of incompressible fluid was used for all streamlines and streamtubes in the past. Using this assumption helps to simplify equations and calculations in streamline simulation. Other assumptions introduced together with incompressible flow are:

1) Streamlines must start in a source and end in a sink
2) Flow rate along each streamline is constant. However, there is no incompressible flow in reality.

In compressible flow, streamlines can start from or end in any grid block (e.g. because of pressure change) but in an incompressible flow, each grid block has constant rate of flow regardless of pressure increace or decreace. When compressible flow moves through grid blocks, the flow rate can change because of pressure difference or other physical factors in one grid block. It means each gird block in a compressible flow can play the role of being sink or a source and change the volume of the flow.

Chapter III

The Pollock Method for Streamline Generation*

In this chapter, we first give detailed description of the Pollock method in 2D computational grids as presented in Pollock (1988). Furthermore, we will extend the streamline simulation method to more general flow problems including gravity, compressibility and time dependency. Finally, we will modify the Pollock method for radial geometries as also discussed in the introduction.

3-1 Mass conservation

The general mass conservation (continuity) equation for a single fluid in a porous medium is:

(3-1) $$\frac{\partial}{\partial t}(\phi\rho) + \nabla.(\rho\bar{v}) = \text{Sources}$$

*) From Pollock, (1998)

where ρ is fluid density and $\bar{v}$ is volumetric flux. If we assume incompressible fluid and rock, this reduces to:

$$(3\text{-}2) \qquad \nabla.(\bar{v}) = 0$$

at any source free point in the medium. Equation (3-2) is the elliptic (time independent) Laplace equation which can be solved for pressure at any location in the medium. Once pressure distribution is known, the velocity field can be calculated from Darcy's Law, which is the base for streamline generation.

3-2 Time of Flight

The variable called Time Of Flight (TOF) was first introduced in Milligan et al. (1951). Physically, it is the time required for a particle that travels along a streamline to reach a location ξ measured curvilinearly along the streamline at ξ from the beginning of the streamline, Fig 3.1.

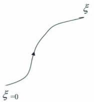


Figure 3.1: Particle Travel Through Streamline

Once the streamlines have been determined from equation (3-2) as explained in section 3-1, the time of flight (TOF) can be determined from:

(3-3) $$t = \int_0^{\xi} \frac{\phi}{u} d\xi$$

where u is the length of the volumetric flux vector; $u = \|\bar{u}\|$.

In multi-phase flow in porous media, the conservation equation for a fluid component i is:

(3-4) $$\frac{\partial}{\partial t}(\rho_i \phi S_i) + \nabla.(\rho_i \bar{u}_i) = \text{Sources of } i$$

where S_i is the fluid saturation. If we assume incompressible fluids, the equation (3-3) transforms equation (3-4) into a one dimensional conservation law:

(3-5) $$\frac{\partial S_i}{\partial t} + \frac{\partial F_i}{\partial \tau} = 0$$

where F_i is the fractional flow of component i. This explains the benefit of streamline simulation over conventional methods: equation (3-5) is one-dimensional and can be solved along individual streamlines using time of flight.

In other words, the streamline method has reduced the 3D flow problem to;

1) Solution of 3D Laplacian
2) Integration of equation (3-5) along one-dimensional streamlines.

It is emphasized that this simplification assumes incompressibility, negligible influence of gravity (all fluids have the same density) and incompressible rock. We will discuss this in more detail later.

Another advantage of the new model of streamline simulation is the consideration of three-dimensional modeling rather than two dimensions. In 2D modeling, the vertical dimension is ignored.

As mentioned the advantage of Pollock's method is competently tracing streamlines in 3D in terms of time of flight (TOF) using a simple analytical calculation method. Darcy's law is used to find velocities across boundaries of grid blocks. It is assumed that the flux is known from the pressure solution and Darcy's law. The exit point of a streamline is calculated. This exit point is the entrance point for the next cell. Assuming linear velocity in each direction, time of flight can be calculated for each cell.

Pollock's method is used for an orthogonal grid, however in reality most reservoirs do not adjust in a Cartesian modeling shape. Thus, the next advancement in the research explored ways to find stream path lines according to the Pollock method but with assuming isotropic transformation that transforms corner point geometry grids (CPG) into unit cubes, (Prevost et al. (2001) or Cordes et al (1994)). Therefore, the Pollock method can also be used for irregular grids.

3-3 The Streamline Generation Procedure in Cartesian Coordinates

3-3-1 Flow Modeling

In finite difference method, finding the velocity at every point is based on an interpolation scheme. The velocity vector is determined at each node of a finite difference grid in the modeled reservoir. In other methods, such as analytical methods, the velocity vector can be found through analytical solutions at every point.

Three kinds of interpolations for this issue may be considered:

A- Step function of interpolation

B- Simple linear interpolation

C- Multi linear interpolation

In -A- the velocity components are constant between grid cells and will only change in the nodes. In -B- the velocity direction changes linearly through the grid cell and independently from other directions. In -C- velocities in three directions are dependent on all three-velocity vectors; therefore each velocity in each direction shall be computed through the other two directions' velocity vectors.

Simple linear interpolation, -B- is used in this research for finding velocity vectors.

From equation (3-1), the mass conservation theory in steady state single fluid incompressible flow can be written as:

$$\frac{\partial(\phi V_x)}{\partial x}+\frac{\partial(\phi V_y)}{\partial y}+\frac{\partial(\phi V_z)}{\partial z}=W \quad (3\text{-}6)$$

V_x, V_y and V_z : Average linear velocity components

ϕ : Porosity

W: Volume rate of water created (Source term) or consumed per unit bulk volume

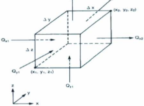


Figure 3.2: Explanation of Fluid Flow in Each Side of Cell, (Pollock (1998))

Figure 3.2 shows one cell and the incoming and outgoing flows diagonally from each of the six faces of the cell. As it is shown in Figure 3.2, x_1, x_2, y_1, y_2, z_1 and z_2 are the coordinates defining the faces of the cell. The average velocity element for each face is found by the volume flow rate divided by cross sectional area for each side and the porosity of the material in the cell.

(3-7)

$$V_{x1} = \frac{Q_{x1}}{(\phi \Delta y \Delta z)}$$

$$V_{x2} = \frac{Q_{x2}}{(\phi \Delta y \Delta z)}$$

$$V_{y1} = \frac{Q_{y1}}{(\phi \Delta x \Delta z)}$$

$$V_{y2} = \frac{Q_{y2}}{(\phi \Delta x \Delta z)}$$

$$V_{z1} = \frac{Q_{z1}}{(\phi \Delta z \Delta y)}$$

$$V_{z2} = \frac{Q_{z2}}{(\phi \Delta z \Delta y)}$$

Q: Volume flow rate across a cell face

$\Delta x, \Delta y, \Delta z$: Length of cell faces

ϕ : Porosity

3-3-2 Discretized Mass Balance Equation

Comparing the above two equations (3-6) & (3-7), a discrete form of mass conservation equation can be written:

(3-8)

$$\frac{(\phi V_{x2} - nV_{x1})}{\Delta x} + \frac{(\phi V_{y2} - nV_{y1})}{\Delta y} + \frac{(\phi V_{z2} - nV_{z1})}{\Delta z} = \frac{Q_s}{\Delta x \Delta y \Delta z}$$

In addition, Darcy's law can be written for each side cell. Substitution of Darcy's law for each of the flow terms in equation (3-8) results in a set of algebraic equation expressed in terms of pressures at nodes located at the cell centers, (McDonald and Harbaugh, (1984)).

The solution of those equations gives the value of flow rate in each node. When the flow rates for each node were calculated, the intercellular flow rates can be computed from Darcy's law using the value of pressure at the nodes points, (Pollock (1988)).

In order to achieve the main target of finding the path lines, the calculation process should be set up to find the elements of velocity vectors at each point in the reservoir.

As it was mentioned before, the main goal is to locate the path lines by using simple linear interpolation. The components of velocity vectors obtained with simple linear interpolation are given by:

$$
\begin{aligned}
V_x &= A_x(x - x_1) + V_{x1} \\
V_y &= A_y(y - y_1) + V_{y1} \\
V_z &= A_z(z - z_1) + V_{z1}
\end{aligned}
\tag{3-9}
$$

where [A_x, A_y, A_z] is velocity gradient within a cell, i.e.

$$
\begin{aligned}
A_x &= \frac{(V_{x2} - V_{x1})}{\Delta x} \\
A_y &= \frac{(V_{y2} - V_{y1})}{\Delta y} \\
A_z &= \frac{(V_{z2} - V_{z1})}{\Delta z}
\end{aligned}
\tag{3-10}
$$

To elaborate, when linear velocity elements shown in equation (3-9) are substituted into equation (3-6), the three derivatives of equation (3-6) will be constant. They are the same as the three terms on the left side of equation (3-8).

Accordingly, linear interpolation of the six cells faces velocity components results in a velocity vector field within the cell that automatically satisfies equation (3-8) at every point inside the cell. This is correct provided that any internal sources or sinks are assumed to be uniformly distributed within the cell, (Pollock (1988)).

The change rate of particle movement can be found as:

(3-11) $$(\frac{dV_x}{dt})_p = (\frac{dV_x}{dx})(\frac{dx}{dt})_p$$

where supscript p is used for "particle";

$(\frac{dx}{dt})_p$ =Velocity

The velocity vector in x -direction is:

(3-12) $$V_{xp} = (\frac{dx}{dt})_p$$

Also, from equation (3-9) the velocity gradient can be found as follows:

(3-13) $$(\frac{dV_x}{dx}) = A_x$$

Consequently, according to equations (3-11), (3-12) and (3-13):

(3-14)
$$(\frac{dV_x}{dt})_p = A_x V_{xp}$$
$$(\frac{dV_y}{dt})_p = A_y V_{yp}$$
$$(\frac{dV_z}{dt})_p = A_z V_{zp}$$

V_{xp} and V_{yp} are the velocity of particle p in x and y direction at (x_p, y_p) location.

Thus equation (3-14), becomes:

(3-15)
$$(\frac{1}{V_{xp}})dV_{xp} = A_x dt$$
$$(\frac{1}{V_{yp}})dV_{yp} = A_y dt$$
$$(\frac{1}{V_{zp}})dV_{zp} = A_z dt$$

Integrating equation (3-15) between t_1 and t_2:

(3-16)
$$\int_{t_1}^{t_2}(\frac{1}{V_{xp}})dV_{xp} = \int_{t_1}^{t_2} A_x dt \Rightarrow \ln(\frac{V_{xp}}{V_{xp}})_{t2,t1} = A_x \Delta t \Rightarrow \ln\left[\frac{V_{xp}(t_2)}{V_{xp}(t_1)}\right] = A_x \Delta t$$
$$\int_{t_1}^{t_2}(\frac{1}{V_{yp}})dV_{yp} = \int_{t_1}^{t_2} A_y dt \Rightarrow \ln(\frac{V_{yp}}{V_{yp}})_{t2,t1} = A_y \Delta t \Rightarrow \ln\left[\frac{V_{yp}(t_2)}{V_{yp}(t_1)}\right] = A_y \Delta t$$
$$\int_{t_1}^{t_2}(\frac{1}{V_{zp}})dV_{zp} = \int_{t_1}^{t_2} A_z dt \Rightarrow \ln(\frac{V_{zp}}{V_{zp}})_{t2,t1} = A_z \Delta t \Rightarrow \ln\left[\frac{V_{zp}(t_2)}{V_{zp}(t_1)}\right] = A_z \Delta t$$

where $\Delta t = t_2 - t_1$.

To find exit point coordinates, $x_p(t_2)$, $y_p(t_2)$ and $z_p(t_2)$ the below procedure should be followed for each direction:

For x direction according to equation series (3-9):

$$V_{xp}(t_2) = A_x(x_p(t_2) - x_1) + V_{x1}$$

$$\Downarrow$$

(3-17) $$x_p(t_2) = \frac{1}{A_x}(V_{xp}(t_2) - V_{x1}) + x_1$$

V_x can be found from equation series (3-16):

$$\int_{t_1}^{t_2}(\frac{1}{V_{xp}})dv_{xp} = \int_{t_1}^{t_2} A_x dt \Rightarrow \ln(\frac{V_{xp}}{V_{xp}})_{t2,t1} = A_x\Delta t \Rightarrow \ln\left[\frac{V_{xp}(t_2)}{V_{xp}(t_1)}\right] = A_x\Delta t$$

$$\Downarrow$$

(3-18) $$\ln\left[\frac{V_{xp}(t_2)}{V_{xp}(t_1)}\right] = A_x\Delta t$$

$$\Downarrow$$

(3-19) $$V_{xp}(t_2) = V_{xp}(t_1)(\exp(A_x\Delta t))$$

Therefore from equations (3-17) and (3-19):

$$x_p(t_2) = \frac{1}{A_x}(V_{xp}(t_2) - V_{x1}) + x_1$$

$$V_{xp}(t_2) = V_{xp}(t_1)(\exp(A_x\Delta t))$$

$$\Rightarrow$$

$$x_p(t_2) = (\frac{1}{A_x})[V_{xp}(t_1)(\exp(A_x\Delta t)) - V_{x1}] + x_1$$

Repeating the above procedure for y and z directions results in:

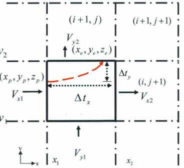
$$x_p(t_2) = x_1 + (\frac{1}{A_x})\left[V_{xp}(t_1)\exp(A_x\Delta t) - V_{x1}\right]$$

$$(3\text{-}20) \qquad y_p(t_2) = y_1 + (\frac{1}{A_y})\left[V_{yp}(t_1)\exp(A_y\Delta t) - V_{y1}\right]$$

$$z_p(t_2) = z_1 + (\frac{1}{A_z})\left[V_{zp}(t_1)\exp(A_z\Delta t) - V_{z1}\right]$$

Therefore, if the velocity at t_1 is known, the location of the particle at time t_2 can be easily found from equation (3-20). This model can be used as an algorithm for finding each path line location in one grid cell and repeating the calculations again for the following next grid cells.

In Figure 3.2 one grid cell is shown with all velocity vectors. V_{x1} and V_{x2} velocities at the x_1 and x_2 faces, respectively. In Figure 3.2, V_{x1}, V_{x2}, V_{y1} and V_{y2} are assumed to be positive.



(x_p, y_p, z_p) : Particle coordinates in three dimensions at the entrance point

(x_e, y_e, z_e) : Particle coordinates at the exit point

Δt_x : Streamline particle time of travel in x direction

Δt_y : Streamline particle time of travel in y direction

V_{x1} : Velocity of Streamline in x direction

V_{y2} : Velocity of Streamline in y direction

Figure 3.3: A Brief Demonstration of Particle Movement, (Pollock (1988))

As mentioned, to find x and y location for each streamline particle for every grid blocks equation series (3-20) is used.

By connecting all enterance and exit points, fluid flow is visualized as streamlines.

As explained before, no particle disappears inside cells. Furthermore, no sinks or sources are assumed to be inside grid cells. Sources or sinks an only be located on the boundaries.

Let (x_p, y_p) be streamline particle coordinate at the entrance point of the first grid cell and V_{xp} be the particle velocity in x direction. Hence V_{xp} can be calculated from equation (3-9). In equation (3-9), V_x is the same as V_{x2} at face x_2 (Figure 3.4).

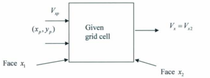


Figure 3.4: Schematic of a Grid cell and Fluid Flow in x Direction

After calculating V_{xp}, the time difference in x and y direction can be easily found from equation (3-18) and equation (3-21) which are resulted from equation (3-16). For y direction results in:

$$\text{(3-21)} \qquad \frac{1}{A_y} \ln\left[\frac{V_{y2}}{V_{yp}}\right] = \Delta t_y$$

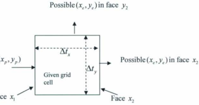


Figure 3.5: Schematic of the Grid Cell and Time of Travel in x and y Directions

After finding Δt_x and Δt_y the time of flight in a grid cell is:

(3-22) $$\Delta t_e = \text{Min } (\Delta t_x, \Delta t_y)$$

Hence, according to above calculation process, this specifies the next target of the particle to enter the next grid block. If Δt_y is less than Δt_x, the particle reaches face y_2 first, leaves the cell across from the face y_2 and enters grid cell $(i+1, j)$. Similarly, if $\Delta t_x < \Delta t_y$, the particle enters cell *(i,j+1)*.

If $\Delta t_x = \Delta t_y$ it can be said that the particle goes exactly through the corner of the cell and will enter to the cell $(i+1, j+1)$.

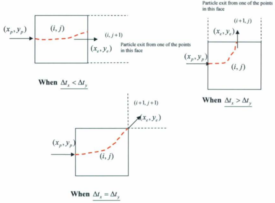


Figure 3.6: Schematic of the Grid Cell and Exit Coordinate Possibilities

We next determine how to find the exit location of particle $P(x_e, y_e)$ using equation (3-20) :

Thus the exit time can be found easily through $t_e = t_p + \Delta t_e$.

This computation series procedure can be reiterated until the mentioned particle reaches the boundary outlet Thus, this kind of iteration shall be generated as one specific algorithm as is shown in Figure 3.7.

According to Figure3.7, this progression of computation for steady state flow can be used for more than two dimensions (x, y).

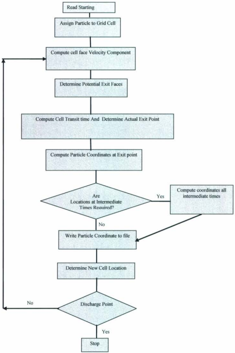


Figure 3.7: Flow Chart- Streamline Modeling (Pollock (1998)).

Chapter IV

Pressure Solution in Cartesian Geometry

This chapter deals with oil streamline numerical modeling in Cartesian coordinates. Each component of the velocity field is approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit in each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point and the exit point is mapped back into physical space to give the entry point into the next cell, and so on.

Note: As an essential assumption in this method, incompressible fluids are assumed and gravity effects are ignored.

4-1 Implementation of Numerical Pressure Solvers in Cartesian Coordinates

At first, mapping the specific finite difference grid blocks onto the unit square (or unit cube in 3D) should be provided for the given reservoir. There is no limitation for choosing the number of grid blocks in this method. It could be different according to the accuracy needed in a study and time available. The Laplace equation is implemented as a numerical pressure solver in Cartesian coordinates. Thus, according to the Laplacian solver one specific pressure is obtained for each grid cell. The following sub- sections show how to find the pressure distribution over a finite difference grid.

4-2 The Laplace Equation In Cartesian Coordinates

The following equation come from material balanced in a source free medium.

(4-1) $\nabla . \vec{u} = 0$

where $\vec{u}$ is volume flux in a source free medium.

For the sake of clarity, we give a simplistic description of the numerical procedure for solving the Laplacian. The description is a compromise between accuracy and simplicity, and for cases with small permeability contrasts between grid blocks, it is reasonably accurate. However, it is emphasized that when solving the general Laplacian adequate upscaling procedures must be used, i.e. arithmetic average of permeabilties for flow parallel to layers and harmonic average for flow perpendicular to layers. For flow in radial geometries, which is the main focous in this work, we will give a rigorous description of the solution of the Laplacian in chapter V. Therefore, it suffices to give a simplistic approach for Cartesian geometries.

The numerical formulation for solving pressure distribution in finite difference grid cells starts with using a simple form of the Laplace equation in two dimensions. Incompressible fluid is assumed, i.e. μ and ρ are constant. This determines the pressure at any point given an appropriate set of boundary conditions.

(4-2) $$\nabla.(\underline{K}\nabla p) = 0$$

$$\Rightarrow$$

(4-3) $$\frac{\partial}{\partial x}[K_x(\frac{\partial p}{\partial x})] + \frac{\partial}{\partial y}[K_y(\frac{\partial p}{\partial y})] = 0$$

If the medium is homogenous then:

(4-4) $$K_x(\frac{\partial^2 p}{\partial x^2}) + K_y(\frac{\partial^2 p}{\partial y^2}) = 0$$

K_x : Permeability in x direction

K_y : Permeability in y direction

p: Pressure

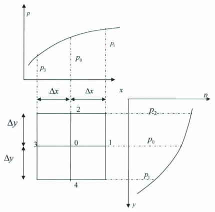


Figure 4.1: Pressure Distribution in a Sample Grid Blocks, (Das (1997))

According to Darcys' law:

(4-5) $$Q = -\frac{KA}{\mu}\frac{p_1 - p_2}{L}$$

Dividing both sides of the equation by the area and using more general notation leads to;

(4-6) $$q = -\frac{K}{\mu}\nabla p$$

where q is the flux (discharge per unit area, with units of length per time, m/s), K is permeability (m^2), μ is viscosity (Pa.s) and ∇p is the dimensionless pressure gradient vector (Pa/m).

In Figure 4.1, p_0, p_1, p_2, p_3 and p_4 are pressures at points *0, 1, 2, 3* and *4* respectingly and Δx and Δy are length and width of each grid block. At the middle point, p_0 can be found through the following calculations using finite difference method.

The following equations can be written for the rate of flow for all four points through the channels shown in Figure 4.1, according to equation (4-6);

$$q_{1-0} = \frac{K_x}{\mu}\frac{p_1 - p_0}{\Delta x}\Delta y.w \qquad \text{from point } 1 \text{ to point } 0$$

(4-7) $$q_{0-3} = \frac{K_x}{\mu}\frac{p_0 - p_3}{\Delta x}\Delta y.w \qquad \text{from point } 0 \text{ to point } 3$$

$$q_{2-0} = \frac{K_y}{\mu}\frac{p_2 - p_0}{\Delta y}\Delta x.w \qquad \text{from point } 2 \text{ to point } 0$$

$$q_{0-4} = \frac{K_y}{\mu}\frac{p_0 - p_4}{\Delta y}\Delta x.w \qquad \text{from point } 0 \text{ to point } 4$$

Here, w is the width of the model perpendicular to x, y -directions.

In Figure 4.1, since the total rate of flow into point 0 is equal to the total rate of flow out of point 0, $q_{in} - q_{out} = 0$

hence,

(4-8) $$(q_{1-0} + q_{2-0}) - (q_{0-3} + q_{0-4}) = 0$$

Substituting equation (4-7) into equation (4-8), we get:

$$\frac{K_x}{\mu}\frac{p_1 - p_0}{\Delta x}\Delta y + \frac{K_y}{\mu}\frac{p_2 - p_0}{\Delta y}\Delta x - \frac{K_x}{\mu}\frac{p_0 - p_3}{\Delta x}\Delta y - \frac{K_y}{\mu}\frac{p_0 - p_4}{\Delta y}\Delta x = 0$$

$\Rightarrow$

$$\frac{K_x\Delta y}{\Delta x}(p_1) - \frac{K_x\Delta y}{\Delta x}(p_0) + \frac{K_y\Delta x}{\Delta y}(p_2) - \frac{K_y\Delta x}{\Delta y}(p_0) - \frac{K_x\Delta y}{\Delta x}(p_0) + \frac{K_x\Delta y}{\Delta x}(p_3) - \frac{K_y\Delta x}{\Delta y}(p_0) + \frac{K_y\Delta x}{\Delta y}(p_4) = 0$$

$\Rightarrow$

$$\frac{K_x\Delta y}{\Delta x}(p_1) + \frac{K_x\Delta y}{\Delta x}(p_3) + \frac{K_y\Delta x}{\Delta y}(p_2) + \frac{K_y\Delta x}{\Delta y}(p_4) = 2(\frac{K_x\Delta y}{\Delta x} + \frac{K_y\Delta x}{\Delta y})p_0$$

$\Rightarrow$

(4-9) $$\frac{\frac{K_x\Delta y}{\Delta x}(p_1 + p_3) + \frac{K_y\Delta x}{\Delta y}(p_2 + p_4)}{2(\frac{K_x\Delta y}{\Delta x} + \frac{K_y\Delta x}{\Delta y})} = p_0$$

Multiply equation (4-9) by $(\Delta y)(\Delta x)$ gives;

(4-10) $$\frac{K_x \Delta y^2 (p_1 + p_3) + K_y \Delta x^2 (p_2 + p_4)}{2(K_x \Delta y^2 + K_y \Delta x^2)} = p_0$$

Equation (4-10) can be used to define a linear system of equations in the unknown pressures. The pressure at the middle point, p_0, will be found from equation (4-10), where K_x and K_y are constant.

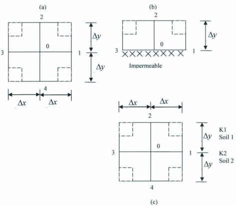


Figure 4.2: Different Case Studies for Solving Pressure Distribution, (Das (1997))

4-2-1 Isotropic Medium

If the medium is isotropic (e.g. Figure 4.2a), the permeabilities in directions x and y are equal.

If $K_x = K_y = K$ and $\Delta x = \Delta y$, equation (4-10) simplifies to

$$(4\text{-}11)\quad p_0 = \frac{(\Delta x)^2 K_x (p_1 + p_3 + p_2 + p_4)}{4((\Delta x)^2 K_x)} = \frac{1}{4}(p_1 + p_3 + p_2 + p_4)$$

4-2-2 Pressure Solving at the Boundary of Impermeable Layers

If the point 0 is located on the boundary of a pervious and impervious layer as shown in Figure 4.2b, equation (4-10) must be modified as follows:

$$(4\text{-}12)\quad \begin{aligned} q_{1-0} &= \frac{K_x}{\mu}\frac{p_1 - p_0}{\Delta x}\frac{\Delta y}{2} \\ q_{0-3} &= \frac{K_x}{\mu}\frac{p_0 - p_3}{\Delta x}\frac{\Delta y}{2} \\ q_{0-2} &= \frac{K_y}{\mu}\frac{p_0 - p_2}{\Delta y}\Delta x \end{aligned}$$

For continuity of flow,

$$(4\text{-}13)\quad q_{1-0} - q_{0-2} - q_{0-3} = 0$$

Substituting equation (4-12) into equation (4-13):

(4-14) $\frac{K_x \Delta y}{2\Delta x \mu}(p_1 - p_0) - \frac{K_y \Delta x}{\Delta y \mu}(p_0 - p_2) - \frac{K_x \Delta y}{2\Delta x \mu}(p_0 - p_3) = 0$

Assuming $\Delta y = \Delta x$, and $K_x = K_y$, it leads to the following equation:

(4-15) $\Delta y = \Delta x \Rightarrow p_0 = \frac{1}{4}(p_1 + p_3 + 2p_2)$

4-2-3 Pressure Solving at the Boundary of Two Different Layers

Equation (4-15) is valid for grid cells in homogenous soils. However, in real fields, each reservoir consists of various layers with diffrent permeabilities. When flow region includes different reservoir layers with different permeabilities (e.g. Figure 4.2c), equation (4-15) must be modified accordingly.

For the case of flow in x direction in Figure 4.2c , the flow region is located equally in layers *1* and *2* with coefficients of permeability of K_1 and K_2, respectingly. According to upscaling laws, when flow direction is parallel to the boundary between two soil layers equivalent coefficient of permeability at x direction can be used as:

(4-16) $K_x = \frac{K_1 + K_2}{2}$

For the case of flow in y- direction in Figure 4.2.c, if we replace soil *2* with soil *1*, and use p_4' instead of p_4 for the pressure at point 4, for the velocity to remain the same,

$$K_1 \frac{p'_4 - p_0}{\Delta y} = K_2 \frac{p_4 - p_0}{\Delta y}$$

(4-17) *or*

$$p'_4 = \frac{K_2}{K_1}(p_4 - p_0) + p_0$$

Thus equation (4-4) can be written as:

$$K_x \frac{\partial^2 p}{\partial x^2} + K_y \frac{\partial^2 p}{\partial y^2} = 0$$

$$\Rightarrow$$

$$\frac{K_1 + K_2}{2} \frac{p_1 + p_3 - 2p_0}{(\Delta x)^2} + K_1 \frac{p_2 + p'_4 - 2p_0}{(\Delta y)^2} = 0$$

$$\Rightarrow$$

$$\frac{K_1 + K_2}{2(\Delta x)^2}(p_1 + p_3 - 2p_0) + \frac{K_1}{(\Delta y)^2}(p_2 + p'_4 - 2p_0) = 0$$

using $p_4^{'}$ from equation (4-17):

$$\frac{K_1 + K_2}{2(\Delta x)^2}(p_1 + p_3 - 2p_0) + \frac{K_1}{(\Delta y)^2}(p_2 + \frac{K_2}{K_1}(p_4 - p_0) + p_0 - 2p_0) = 0$$

$$\Rightarrow$$

$$\frac{K_1 + K_2}{2(\Delta x)^2}(p_1 + p_3) - \frac{K_1 + K_2}{(\Delta x)^2}(p_0) + \frac{K_1}{(\Delta y)^2}(p_2) + \frac{K_2}{(\Delta y)^2}(p_4) - \frac{K_2}{(\Delta y)^2}(p_0) - \frac{K_1}{(\Delta y)^2}(p_0) = 0$$

$\Rightarrow$

$$\frac{K_1+K_2}{2(\Delta x)^2}(p_1+p_3)+\frac{K_1}{(\Delta y)^2}(p_2)+\frac{K_2}{(\Delta y)^2}(p_4)=(\frac{K_1+K_2}{(\Delta x)^2}+\frac{K_2}{(\Delta y)^2}+\frac{K_1}{(\Delta y)^2})p_0$$

Multipling by $(2\Delta x^2 \Delta y^2)$;

$$(\Delta y)^2(K_1+K_2)(p_1+p_3)+2K_1(\Delta x)^2(p_2)+2K_2(\Delta x)^2(p_4)=2(K_1+K_2)(\Delta y^2+\Delta x^2)p_0$$

Solving for p_0, the general equation for this case would be:

(4-18) $$p_0=\frac{(\Delta y)^2(K_1+K_2)(p_1+p_3)+2K_1(\Delta x)^2(p_2)+2K_2(\Delta x)^2(p_4)}{2(K_1+K_2)(\Delta y^2+\Delta x^2)}$$

If taking $\Delta y=\Delta x$;

$$p_0=\frac{(K_1+K_2)(p_1+p_3)+2K_1(p_2)+2K_2(p_4)}{4(K_1+K_2)}$$

$\Rightarrow$

(4-19) $$p_0=\frac{1}{4}(p_1+p_3)+\frac{K_1}{2(K_1+K_2)}(p_2)+\frac{K_2}{2(K_1+K_2)}(p_4)$$

Therefore having the relevant equation according to the grid cell and boundary conditions in the assumed finite difference grid, the pressure distribution for the whole block will be found.

4-3 Velocity

After finding pressures for all grid cells, the volumetric flux can be found according to Darcy's law, for each grid cell.

$$Q = K\frac{A}{\rho g}(\frac{p_1 - p_2}{L} + \rho g)$$

(4-20) $$\Rightarrow$$

$$Q = K\frac{A}{\mu}(\frac{p_1 - p_2}{L} + \rho g)$$

where:

μ : Fluid viscosity

ρ : Fluid density

K : Permeability of the soil and

A : Cross section of the cell

(4-21) $$u = \frac{Q}{A} \Rightarrow u = \frac{K}{\mu}(\frac{p_1 - p_2}{L} + \rho g)$$

The differential form of Darcy's Law is:

(4-22) $$u = -\frac{K}{\mu}(\frac{\partial p}{\partial x} + \rho g \sin\theta)$$

Here, L is flow direction and θ is the angle between the positive x - axis and positive flow direction measured counter clockwise. Therefore, according to all pressures derived from the Laplacian and Darcy's Law, the entrance velocity and exit velocity will be found for each grid cell.

After substituting all relevant coefficients in Darcy's Law, volumetric flux or Darcy's velocity in x direction will be found either as entrance velocity in the grid or exit velocity from the grid cell. Consequently after finding pressure and velocity for each grid cell, time of travel in x direction will be found easily according to equations (3-15) and (3-16).

For y direction, all the above calculations process is repeated. Then, according to equation (3-16), the time of travel for y direction will be found as well. In the next step, the time of flight is calculated according to equation (3-22).

Finally, all required coefficients are substituted in equation (3-20) thus x_e and y_e as exit point for the streamline particle in one gird block will be found.

These calculation series can be reiterated until the assumed particle reaches to the boundary. Hence all x_e and y_e (exit coordinates) are assumed the entrance points for the next grid cell and the calculation procedure can be repeated.

After all, from the beginning until the last boundary place, there are bench-marks for each grid cell as entrance points and exit points for streamline flow. Easily they can connect together like a chain and produce one prolonged line that shows one streamline in the given reservoir. It visualizes streamline behavior according to various geological and process factors and also physical boundary conditions.

Chapter V

Pressure solution in Radial Geometries*

5-1 Polar Coordinate System

Let x, y and z be the Cartesian coordinate system aligned with the principal permeability directions with permabilities K_x, K_y and K_z respectively. Assume a well is parallel to the z-axis. Consider a cylindrical cross section perpendicular to the well, Figure 5.1.

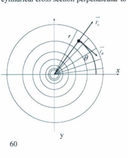


Figure 5.1: Radial Geometry

*) From T. Johansen, (2009)

The Polar representation of point (x, y) , (r, θ) is:

$$
\begin{aligned}
& x = r\cos\theta \\
& y = r\sin\theta \\
(5\text{-}1) \qquad & r = \sqrt{x^2 + y^2} \\
& \theta = \arctan(\frac{y}{x})
\end{aligned}
$$

The gradient of a vector in Cartesian coordinates is:

$$
(5\text{-}2) \qquad \nabla = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right]
$$

Using the chain rule for differentiation:

$$
(5\text{-}3) \qquad
\begin{aligned}
\frac{\partial}{\partial x} &= \frac{\partial}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial x} \\
\frac{\partial}{\partial y} &= \frac{\partial}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial y}
\end{aligned}
$$

According to equation (5-1), in x direction, it is calculated:

$x = r\cos\theta \Rightarrow$

$y = r\sin\theta \Rightarrow$

$$
r = \sqrt{x^2 + y^2} \Rightarrow \frac{\partial r}{\partial x} = \frac{2x}{2\sqrt{x^2 + y^2}} = \frac{x}{r} = \frac{r\cos\theta}{r} = \cos\theta
$$

$$
(5\text{-}4) \qquad \frac{\partial r}{\partial y} = \frac{2y}{2\sqrt{x^2 + y^2}} = \frac{y}{r} = \frac{r\sin\theta}{r} = \sin\theta
$$

$$
(5\text{-}5) \qquad \frac{\partial \theta}{\partial x} = -\frac{1}{r}\sin\theta
$$

Consequently, the same process for y direction should be done:

$$\frac{\partial \theta}{\partial y} = \frac{1}{r}\cos\theta$$

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial r}\cos\theta - \frac{1}{r}\sin\theta\frac{\partial}{\partial \theta} \tag{5-6}$$

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial r}\sin\theta + \frac{1}{r}\cos\theta\frac{\partial}{\partial \theta}$$

5-2 Gradient in Polar Coordinate

Hence, the gradient in Polar coordinates becomes:

$$\nabla = \left[\frac{\partial}{\partial r}\cos\theta - \frac{1}{r}\sin\theta\frac{\partial}{\partial \theta}, \frac{\partial}{\partial r}\sin\theta + \frac{1}{r}\cos\theta\frac{\partial}{\partial \theta}\right] \tag{5-7}$$

The components of gradient in the direction of the unit vectors $\vec{e_r}$, $\vec{e_\theta}$ in Figure 5.1, respectively, are:

$$\vec{e_r} = [\cos\theta, \sin\theta] \tag{5-8}$$
$$\vec{e_\theta} = [-\sin\theta, \cos\theta]$$

5-2-1 Gradient in *r* Direction

Therefore the gradient in r direction is:

$$\nabla_r = \vec{e_r} \bullet \nabla = [\cos\theta, \sin\theta] \bullet \left[\frac{\partial}{\partial r}\cos\theta - \frac{1}{r}\sin\theta\frac{\partial}{\partial \theta}, \frac{\partial}{\partial r}\sin\theta + \frac{1}{r}\cos\theta\frac{\partial}{\partial \theta}\right]$$

$$\nabla_r = \vec{e_r} \bullet \nabla = \left[\frac{\partial}{\partial r}\cos^2\theta - \frac{1}{r}\sin\theta\cos\theta\frac{\partial}{\partial \theta} + \frac{\partial}{\partial r}\sin^2\theta + \frac{1}{r}\sin\theta\cos\theta\frac{\partial}{\partial \theta}\right]$$

(5-9) $$\frac{\partial}{\partial r} = \left[\frac{\partial}{\partial r}(\cos^2\theta + \sin^2\theta) - \frac{1}{r}\sin\theta\cos\theta\frac{\partial}{\partial\theta} + \frac{1}{r}\sin\theta\cos\theta\frac{\partial}{\partial\theta}\right]$$

5-2-2 Gradient in θ Direction

The gradient in θ direction will be:

(5-10) $$\nabla_\theta = \vec{e_\theta}\bullet\nabla = \left[-\sin\theta, \cos\theta\right]\bullet\left[\frac{\partial}{\partial r}\cos\theta - \frac{1}{r}\sin\theta\frac{\partial}{\partial\theta}, \frac{\partial}{\partial r}\sin\theta + \frac{1}{r}\cos\theta\frac{\partial}{\partial\theta}\right]$$

$$\nabla_\theta = \vec{e_\theta}\bullet\nabla = -\sin\theta\cos\theta\frac{\partial}{\partial r} + \frac{1}{r}\sin^2\theta\frac{\partial}{\partial\theta} + \frac{\partial}{\partial r}\cos\theta\sin\theta + \frac{1}{r}\cos^2\theta\frac{\partial}{\partial\theta}$$

$\Rightarrow$

$$\frac{1}{r}\frac{\partial}{\partial\theta} = \frac{\partial}{\partial r}(\sin\theta\cos\theta - \sin\theta\cos\theta) + \frac{1}{r}\frac{\partial}{\partial\theta}(\sin^2\theta + \cos^2\theta)$$

Hence:

$$\nabla_r = \frac{\partial}{\partial r}$$
$$\nabla_\theta = \frac{1}{r}\frac{\partial}{\partial\theta}$$

Combining equation (5-9) and equation (5-10),

(5-11) $$\nabla_{r,\theta} = \left[\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial\theta}\right]$$

The divergence of a vector $\vec{u}$ in Cartesian coordinates is:

(5-12) $$Div\vec{u} = \nabla\bullet\vec{u} = \left[\frac{\partial u_x}{\partial x}, \frac{\partial u_y}{\partial y}\right]$$

5-3 Velocity in Polar Direction

Let u_r, u_θ be the coordinate of $\vec{u}$ in the coordinate system e_r, e_θ,

(5-13) $$\vec{u} = [u_r, u_\theta]$$

Note that:

(5-14)
$$\vec{e_r} = [\cos\theta, \sin\theta] \Rightarrow \frac{\partial \vec{e_r}}{\partial\theta} = [-\sin\theta, \cos\theta] = \vec{e_\theta}$$
$$\frac{\partial \vec{e_r}}{\partial r} = 0$$
$$\vec{e_\theta} = [-\sin\theta, \cos\theta] \Rightarrow \frac{\partial \vec{e_\theta}}{\partial\theta} = [-\cos\theta, -\sin\theta] = -\vec{e_r}$$
$$\frac{\partial \vec{e_\theta}}{\partial r} = 0$$

Therefore, the velocity in Polar coordinate are:

(5-15)
$$u_r = \vec{e_r} \bullet \vec{u}$$
$$u_\theta = \vec{e_\theta} \bullet \vec{u}$$

Therefore, the divergence in Polar coordinates will be:

$$\nabla_{r,\theta} \bullet \vec{u} = \left[\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial\theta}\right] \bullet \vec{u}$$

(5-16) $$\nabla_{r,\theta} \bullet \vec{u} = \left[\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial\theta}\right] \bullet \vec{u} = \left[\vec{e_r}(\frac{\partial}{\partial r}) + \vec{e_\theta}\frac{1}{r}(\frac{\partial}{\partial\theta})\right] \bullet \left[u_r\vec{e_r} + u_\theta\vec{e_\theta}\right]$$

$$\nabla_{r,\theta} \bullet \vec{u} = \left[\vec{e_r}\frac{\partial}{\partial r}(u_r\vec{e_r}) + \vec{e_r}\frac{\partial}{\partial r}(u_\theta\vec{e_\theta}) + \vec{e_\theta}\frac{1}{r}\frac{\partial}{\partial\theta}(u_r\vec{e_r}) + \vec{e_\theta}\frac{1}{r}\frac{\partial}{\partial\theta}(u_\theta\vec{e_\theta})\right]$$

$\Rightarrow$

$$\nabla_{r,\theta} \bullet \vec{u} = \vec{e_r} \bullet \vec{e_r} \frac{\partial u_r}{\partial r} + \vec{e_r} u_r \frac{\partial \vec{e_r}}{\partial r} + \vec{e_r} \bullet \vec{e_\theta} \frac{\partial u_\theta}{\partial r} + \vec{e_r} u_\theta \frac{\partial \vec{e_\theta}}{\partial r} + \vec{e_r} \bullet \vec{e_\theta} \frac{1}{r} \frac{\partial u_r}{\partial \theta} + \vec{e_\theta} u_r \frac{1}{r} \frac{\partial \vec{e_r}}{\partial \theta} + \vec{e_\theta} \bullet \vec{e_\theta} \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \vec{e_\theta} u_\theta \frac{1}{r} \frac{\partial \vec{e_\theta}}{\partial \theta}$$

Hence, according to equation (5-14), the above equation (5-15) can be written as:

$$\nabla_{r,\theta} \bullet \vec{u} = \frac{\partial u_r}{\partial r} + 0 + 0 + 0 + 0 + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} - 0 = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = \frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta}$$

which gives;

(5-17) $$\nabla_{r,\theta} \bullet \vec{u} = \frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{\partial (u_\theta)}{\partial \theta}$$

5-4 Darcy's Law

Darcy's Law in Cartesian coordinates using principal permeability directions is:

(5-18) $$\vec{u} = -\frac{1}{\mu} \underline{K} . \nabla p$$

Where since coordinate axes are parallel to the principal permeability directions;

(5-19) $$\underline{K} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix}$$

The explicit expressions for the flux vector components u_r and u_θ are as follows:

According to equations (5-14), (5-15) and (5-18):

(5-20) $$u_r = \vec{e_r} . \vec{u} = -\frac{1}{\mu}\left[\cos\theta, \sin\theta\right] \bullet \left[K_x \frac{\partial p}{\partial x}, K_y \frac{\partial p}{\partial y} \right]$$

According to equation (5-3):

$$(5\text{-}21)\qquad \begin{aligned}\frac{\partial}{\partial x}&=\frac{\partial}{\partial r}\frac{\partial r}{\partial x}+\frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial x}\\ \frac{\partial}{\partial y}&=\frac{\partial}{\partial r}\frac{\partial r}{\partial y}+\frac{\partial}{\partial \theta}\frac{\partial \theta}{\partial y}\end{aligned}\quad\Rightarrow\quad \begin{aligned}\frac{\partial}{\partial x}&=\frac{\partial}{\partial r}\cos\theta-\frac{1}{r}\sin\theta\frac{\partial}{\partial \theta}\\ \frac{\partial}{\partial y}&=\frac{\partial}{\partial r}\sin\theta+\frac{1}{r}\cos\theta\frac{\partial}{\partial \theta}\end{aligned}\quad\Rightarrow$$

$$\begin{aligned}\frac{\partial p}{\partial x}&=\frac{\partial p}{\partial r}\cos\theta-\frac{1}{r}\sin\theta\frac{\partial p}{\partial \theta}\\ \frac{\partial p}{\partial y}&=\frac{\partial p}{\partial r}\sin\theta+\frac{1}{r}\cos\theta\frac{\partial p}{\partial \theta}\end{aligned}$$

5-4-1- Velocity in *r* direction

The velocity in *r* direction can be obtained from following equations:

$$u_r=-\frac{1}{\mu}\left[\cos\theta,\sin\theta\right]\bullet\left[\frac{\partial p}{\partial r}\times K_x\cos\theta-\frac{\partial p}{\partial \theta}\times\frac{K_x\sin\theta}{r},\frac{\partial p}{\partial r}\times K_y\sin\theta+\frac{\partial p}{\partial \theta}\times\frac{K_y\cos\theta}{r}\right]$$

$\Rightarrow$

$$u_r=-\frac{1}{\mu}\left[\frac{\partial p}{\partial r}\times K_x\cos^2\theta-\frac{\partial p}{\partial \theta}\times\frac{K_x\cos\theta\sin\theta}{r}+\frac{\partial p}{\partial r}\times K_y\sin^2\theta+\frac{\partial p}{\partial \theta}\times\frac{K_y\sin\theta\cos\theta}{r}\right]$$

$$(5\text{-}22)\qquad u_r=-\frac{1}{\mu}\left[\frac{\partial p}{\partial r}(K_x\cos^2\theta+K_y\sin^2\theta)+\frac{1}{r}\frac{\partial p}{\partial \theta}\times\sin\theta\cos\theta(K_y-K_x)\right]$$

5-4-2 Velocity in θ Direction

The velocity in θ direction can be obtained from below equations:

$$u_\theta = \vec{e}_\theta \bullet \vec{u} = -\frac{1}{\mu}[-\sin\theta, \cos\theta] \bullet \left[K_x \frac{\partial p}{\partial x}, K_y \frac{\partial p}{\partial y}\right]$$

$$u_\theta = \vec{e}_\theta \bullet \vec{u} = -\frac{1}{\mu}[-\sin\theta, \cos\theta] \bullet \left[\frac{\partial p}{\partial r} \times K_x \cos\theta - \frac{\partial p}{\partial \theta} \times \frac{K_x \sin\theta}{r}, \frac{\partial p}{\partial r} \times K_y \sin\theta + \frac{\partial p}{\partial \theta} \times \frac{K_y \cos\theta}{r}\right]$$

$$u_\theta = -\frac{1}{\mu}\left[-\frac{\partial p}{\partial r} \times K_x \sin\theta\cos\theta + \frac{\partial p}{\partial \theta} \times \frac{K_x \sin^2\theta}{r} + \frac{\partial p}{\partial r} \times K_y \cos\theta\sin\theta + \frac{\partial p}{\partial \theta} \times \frac{K_y \cos^2\theta}{r}\right]$$

$$u_\theta = -\frac{1}{\mu}\left[\frac{\partial p}{\partial r} \times (K_y \cos\theta\sin\theta - K_x \sin\theta\cos\theta) + \frac{\partial p}{\partial \theta}\left(\frac{K_x \sin^2\theta}{r} + \frac{K_y \cos^2\theta}{r}\right)\right]$$

$\Rightarrow$

$$u_\theta = -\frac{1}{\mu}\left[\frac{\partial p}{\partial r} \times \cos\theta\sin\theta(K_y - K_x) + \frac{1}{r}\frac{\partial p}{\partial \theta}(K_x \sin^2\theta + K_y \cos^2\theta)\right] \tag{5-23}$$

5-5 Permeability representated in Polar Coordinates

Introducing the following derived quantities:

$$\begin{aligned} K_r &= K_x \cos^2\theta + K_y \sin^2\theta \\ K_t &= K_x \sin^2\theta + K_y \cos^2\theta \\ K_\theta &= (K_y - K_x)\sin\theta\cos\theta \end{aligned} \tag{5-24}$$

According to the above equation series, equations (5-22), (5-23) can be written as:

$$\begin{aligned} u_r &= -\frac{1}{\mu}\left[\frac{\partial p}{\partial r}(K_x \cos^2\theta + K_y \sin^2\theta) + \frac{1}{r}\frac{\partial p}{\partial \theta} \times \sin\theta\cos\theta(K_y - K_x)\right] = -\frac{1}{\mu}\left(K_r \frac{\partial p}{\partial r} + \frac{1}{r} K_\theta \frac{\partial p}{\partial \theta}\right) \\ u_\theta &= -\frac{1}{\mu}\left[\frac{\partial p}{\partial r} \times \cos\theta\sin\theta(K_y - K_x) + \frac{1}{r}\frac{\partial p}{\partial \theta}(K_x \sin^2\theta + K_y \cos^2\theta)\right] = -\frac{1}{\mu}\left(K_\theta \frac{\partial p}{\partial r} + \frac{1}{r} K_t \frac{\partial p}{\partial \theta}\right) \end{aligned} \tag{5-25}$$

Therefore:

$$(5\text{-}26)\qquad \begin{aligned} u_r &= -\frac{1}{\mu}(K_r\frac{\partial p}{\partial r}+\frac{1}{r}K_\theta\frac{\partial p}{\partial \theta}) \\ u_\theta &= -\frac{1}{\mu}(K_\theta\frac{\partial p}{\partial r}+\frac{1}{r}K_t\frac{\partial p}{\partial \theta}) \end{aligned} \qquad \Rightarrow u=\left[u_r,u_\theta\right]=-\frac{1}{\mu}\begin{pmatrix} K_r & K_\theta \\ K_\theta & K_t \end{pmatrix}\begin{pmatrix} \frac{\partial p}{\partial r} \\ \frac{1}{r}\frac{\partial p}{\partial \theta} \end{pmatrix}$$

Recalling equation (5-18), it turns into the following equation:

$$(5\text{-}27)\qquad \vec{u}=-\frac{1}{\mu}\underline{K}\bullet\nabla_{r,\theta}p=-\frac{1}{\mu}\begin{pmatrix} K_r & K_\theta \\ K_\theta & K_t \end{pmatrix}\bullet\begin{pmatrix} \frac{\partial p}{\partial r} \\ \frac{1}{r}\frac{\partial p}{\partial \theta} \end{pmatrix}$$

This is Darcy's law in Polar coordinates.

Note that K_r and K_θ are directional permeabilities in the radial and tangential directions, respectively.

5-6 The Laplace Equation in Polar Coordinates

The Laplacian in Polar coordinates is:

$\nabla_{r,\theta}.\vec{u}=0$

Using equations (5-15), (5-25) and (5-26);

$$(5\text{-}28)\quad \nabla_{r,\theta}.\vec{u}=\frac{1}{r}\frac{\partial}{\partial r}(ru_r)+\frac{1}{r}\frac{\partial}{\partial \theta}(u_\theta)=\frac{1}{r}\frac{\partial}{\partial r}(rK_r\frac{\partial p}{\partial r}+K_\theta\frac{\partial p}{\partial \theta})+\frac{1}{r}\frac{\partial}{\partial \theta}(K_\theta\frac{\partial p}{\partial r}+\frac{1}{r}K_t\frac{\partial p}{\partial \theta})$$

Hence, the general Laplacian in Polar coordinates becomes:

$$(5\text{-}29) \quad \frac{1}{r}\frac{\partial}{\partial r}(rK_r\frac{\partial p}{\partial r})+\frac{1}{r}\frac{\partial}{\partial r}(K_\theta\frac{\partial p}{\partial \theta})+\frac{1}{r}\frac{\partial}{\partial \theta}(K_\theta\frac{\partial p}{\partial r})+\frac{1}{r^2}\frac{\partial}{\partial \theta}(K_t\frac{\partial p}{\partial \theta})=0$$

This is the general Laplacian in Polar coordinates for the case when the Cartesian coordinate system aligns with the principal permeability direction.

5-6-1 Laplacian in Isotropic Medium Case

For an isotropic medium, $K_r = K_t = K$ and $K_\theta = 0$ for this case, the Laplacian becomes:

$$(5\text{-}30) \quad \frac{1}{r}\frac{\partial}{\partial r}(rK\frac{\partial p}{\partial r})+\frac{1}{r^2}\frac{\partial}{\partial \theta}(K\frac{\partial p}{\partial \theta})=0$$

5-6-2 Laplacian in Homogenous Medium Case

For a homogeneous medium, permeabilities in equation (5-29) can be placed outside differentiation:

$$(5\text{-}31) \quad K_r\frac{\partial^2 p}{\partial r^2}+\frac{K_r}{r}\frac{\partial p}{\partial r}+\frac{2K_\theta}{r}\frac{\partial^2 p}{\partial r\partial \theta}+\frac{K_t}{r^2}\frac{\partial^2 p}{\partial \theta^2}=0$$

5-6-3 Laplacian in Homogenous and Isotropic Case

For the homogenous and isotropic case, equation (5-29) reduces to:

$$(5\text{-}32) \quad \frac{\partial^2 p}{\partial r^2}+\frac{1}{r}\frac{\partial p}{\partial r}+\frac{1}{r^2}\frac{\partial^2 p}{\partial \theta^2}=0$$

5-7 General Laplacian in 3D Polar Coordinate System

In 3D case studies, it is straightforward to add the z direction to the 2D description since the z direction is a principle permeability direction. Using the unit vectors,

$$\text{(5-33)} \qquad \begin{aligned} \vec{e_r} &= [\cos\theta, \sin\theta, 0] \\ \vec{e_\theta} &= [-\sin\theta, \cos\theta, 0] \\ \vec{e_z} &= [0,0,1] \end{aligned}$$

Darcy's law will be:

$$\text{(5-34)} \qquad \vec{u} = -\frac{1}{\mu}\begin{pmatrix} K_r & K_\theta & 0 \\ K_\theta & K_t & 0 \\ 0 & 0 & K_z \end{pmatrix} \bullet \begin{pmatrix} \dfrac{\partial p}{\partial r} \\ \dfrac{1}{r}\dfrac{\partial p}{\partial \theta} \\ \dfrac{\partial p}{\partial z} \end{pmatrix}$$

Thus the general Laplacian would be:

$$\text{(5-35)} \qquad \frac{1}{r}\frac{\partial}{\partial r}(rK_r\frac{\partial p}{\partial r}) + \frac{1}{r}\frac{\partial}{\partial r}(K_\theta\frac{\partial p}{\partial \theta}) + \frac{1}{r}\frac{\partial}{\partial \theta}(K_\theta\frac{\partial p}{\partial r}) + \frac{1}{r^2}\frac{\partial}{\partial \theta}(K_t\frac{\partial p}{\partial \theta}) + \frac{\partial}{\partial z}(K_z\frac{\partial p}{\partial z}) = 0$$

5-8 The General Procedures in Streamline Calculations in a Radial Geometry

The velocities u_r and u_θ given by equation (5-26) are used in the Pollock method to generate streamlines. These velocities can be approximated once the Laplacian is solved for pressure on a radial grid.

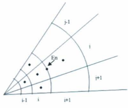


Figure 5.2: Radial Grid

According to Figure 5.2, a grid block (i, j) is considered and an entrance point E_n for the streamline is assumed. The goal is to find the exit point, which must be located on one of the four faces of the grid blocks. Thus, after finding the pressure distribution throughout the grid blocks, the velocities u_r and u_θ on each of the faces of grid (i, j) are calculated. They are estimated using the pressures in the nodes shown in Figure 5.2. In addition, correct upscaling procedures for permeability must be used in the calculation procedure. In order to calculate the exit point, the velocity field inside of (i, j) is needed as well.

5-8-1 Velocity Formula Assumption in θ Direction

The general formula of velocity in the θ direction is assumed linear, and is given as:

(5-36) $$u_\theta(\theta) = a\theta + b$$

The constants a and b are determined to match the known velocities on the faces of grid (i, j).

(5-37) $$a = \frac{u_{\theta 2}(\theta) - u_{\theta 1}(\theta)}{(\theta_2 - \theta_1)}$$

(5-38) $$b = u_{\theta 1}(\theta) - \theta_1 (\frac{u_{\theta 2}(\theta) - u_{\theta 1}(\theta)}{(\theta_2 - \theta_1)})$$

5-8-2 Velocity Formula Assumption in r Direction

General formula of velocity in the r direction will be:

(5-39) $$u_r(r) = \frac{a}{r} + b$$

The constants a and b are found according to known boundary conditions which are coming from case study substituted in equation (5-39);

(5-40) $$a = \frac{u_{r2}(r) - u_{r1}(r)}{(\frac{1}{r_2} - \frac{1}{r_1})}$$

(5-41) $$b = u_{r1}(r) - \frac{1}{r_1}(\frac{u_{r2}(r) - u_{r1}(r)}{(\frac{1}{r_2} - \frac{1}{r_1})})$$

This is consistent with a radial flow situation with an angular "leak off".

This approach is novel and differs from the Cartesian case, as it is a non-linear relationship. Consequently, the formulas for exit point and time of flight calculation are also novel. These formulas are derived next.

5-8-3 Finding Time of Flight for Streamline Movement in Grid Blocks

Travel Time in r Direction

In Figure 5.2, the time needed for a particle to travel with the velocity $u_r(r)$ from the entrance point to the opposite angular face is:

$$t_r = \int_{rEn}^{rEx} \frac{dr}{\frac{a}{r}+b} = \frac{1}{b}\int_{rEn}^{rEx} \frac{brdr}{a+br} = \frac{1}{b}\left[\int_{rEn}^{rEx} \frac{(a+br)dr}{a+br} - \int_{rEn}^{rEx} \frac{adr}{a+br}\right] =$$

$\Rightarrow$

$$t_r = \frac{1}{b}\left[\int_{rEn}^{rEx} dr - \int_{rEn}^{rEx} \frac{adr}{a+br}\right] = \left[\frac{bx - a\log(a+bx)}{b^2}\right]_{rEn}^{rEx} = \frac{1}{b^2}[(b(r_{Ex}) - a\log(a+b(r_{Ex}))) - (b(r_{En}) - a\log(a+b(r_{En})))] =$$

$\Rightarrow$

$$t_r = \frac{1}{b^2}[b(r_{Ex} - r_{En}) - a[\log(a+b(r_{Ex})) - \log(a+b(r_{En}))]] = \frac{1}{b^2}[b(r_{Ex} - r_{En}) - a(l\operatorname{og}(\frac{a+b(r_{Ex})}{a+b(r_{En})}))]$$

$\Downarrow$

(5-42) $$t_r = \frac{1}{b^2}(b(r_{Ex} - r_{En}) - a(l\operatorname{og}(\frac{a+b(r_{Ex})}{a+b(r_{En})})))$$

a and b can be substituted from equations (5-40) and (5-41), to the t_r equation (5-42).

Travel Time in θ Direction

Similarly, the time needed for the particle to travel with velocity $u_\theta(\theta)$ from the entrance point to the opposite face in radial direction is:

$$t_\theta = \int_{\theta En}^{\theta Ex} \frac{d\theta}{a\theta + b} = \left[\frac{\log(a\theta + b)}{a}\right]_{\theta_{En}}^{\theta_{Ex}} = (\frac{\log(a\theta_{Ex} + b)}{a} - \frac{\log(a\theta_{En} + b)}{a}) = \frac{1}{a}\log(\frac{a\theta_{Ex} + b}{a\theta_{En} + b})$$

$\Rightarrow$

(5-43) $$t_\theta = \frac{1}{a}\log(\frac{a\theta_{Ex} + b}{a\theta_{En} + b})$$

a and b can be substituted from equation (5-37) and equation (5-38), to the t_θ equation (5-43).

Time of Flight

The time for the particle to travel from the entrance to the exit face is then,

(5-44) $$t = Min(t_r, t_\theta)$$

This is the time of flight as discussed before. The following equations determine r_{exit} and θ_{exit} :

(5-45) $$r_{exit}(i,j) = r_{particle}(i+\frac{1}{2},j) - (ur_{particle}(i+\frac{1}{2},j) \times T_{\min})$$

(5-46) $$\theta_{exit}(i,j) = \theta_{particle}(i,j) - [(\frac{u_\theta(i+\frac{1}{2},j)}{r(i+\frac{1}{2},j)}) \times T_{\min}]$$

5-9 Discretization

In constructing the grid in Figure 5.2 the nodes are chosen such that the pressure drop between nodes in ring number i and ring number $(i+1)$ is the same for all $i=1,2,3...,N$ for the special case when the cylinder is homogenous and isotropic.

5-10 Radius Equation

Grid blocks are ring tragments as shown in Figure 5.2, The relation between the radius of consecutive rings is:

$$r_{i+1}=r_i M=r_w M^{i+1}$$

where M is constant and $r_0=r_w=$ well bore radius.

If r_e is the drainage radius,

$$r_N=r_e=r_w M^N \Rightarrow M=(\frac{r_e}{r_w})^{\frac{1}{N}}$$

Therefore:

$$r_{i+1}=r_w(\frac{r_e}{r_w})^{\frac{i}{N}} \tag{5-47}$$

5-11 Flow Rate Equation

The flow rate between block i and $(i+1)$ can be calculated using Darcy's law.

$$(5\text{-}48) \qquad q_{i+1/2} = \frac{2\pi K r_{i+1/2} \Delta z}{\mu} \frac{dp}{dr} = \frac{2\pi K r_{i+1/2} \Delta z}{\mu} \frac{p_{i+1} - p_i}{r_{i+1} - r_i}$$

where Δz is the length of the well segment. Integrating equation (5-48) we get:

$$(5\text{-}49) \qquad q_{i+1/2} = \frac{2\pi K \Delta z}{\mu} \frac{(p_{i+1} - p_i)}{\ln(\frac{r_{i+1}}{r_i})}$$

Here, $r_{i+\frac{1}{2}}$ is the radius of the ring between blocks i and $(i+1)$. Using these two expressions for $q_{i+\frac{1}{2}}$ we find:

$$(5\text{-}50) \qquad r_{i+1/2} = \frac{(r_{i+1} - r_i)}{\ln(\frac{r_{i+1}}{r_i})}$$

A second consistent definition of $r_{i+\frac{1}{2}}$ is obtained if we use:

$$q_{i+1/2} = \frac{2\pi K \Delta z}{\mu} \frac{(p_{i+1} - p_{i+1/2})}{\ln(\frac{r_{i+1}}{r_i})} = \frac{2\pi K \Delta z}{\mu} \frac{(p_{i+1/2} - p_i)}{\ln(\frac{r_{i+1/2}}{r_i})}$$

Demanding $p_{i+1} - p_{i+\frac{1}{2}} = p_{i+\frac{1}{2}} - p_i$ we find:

$$(5\text{-}51) \qquad r_{i+1/2} = \sqrt{r_i r_{i+1}}$$

5-12 Permeability Equation

5-12-1 Radial Mobility

Permeability is considered to be a function of the radius:

$K = K(r)$

This is reasonable because well bore damage caused by mud invasion is constant in the angular direction and various radially.

Then flow rate equation can be presented as:

(5-52) $$Q = \frac{2\pi h}{\mu \int_{r_w}^{r_e} \frac{dr}{K(r) r}} (P_e - P_w)$$

Consequently, following formula is used for permeability:

(5-53) $$\overline{K} = \frac{\log(\frac{r_e}{r_w})}{\int_{r_w}^{r_e} \frac{dr}{K(r) r}} = \frac{\log(\frac{r_e}{r_w})}{\sum_{i=1}^{N} \frac{1}{K_i} \log(\frac{r_i}{r_{i-1}})}$$

According to equation (5-52) and using transmissibility λ, i. e. $\lambda = \frac{K}{\mu}$

The upscaled radial transmissibility will be:

(5-54) $$\lambda_{i+1/2}^{r} = \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r} \ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^r} \ln(\frac{r_{i+1}}{r_{i+1/2}})}$$

(5-55) $$\lambda^r_{i-1/2} = \frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda^r_i}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda^r_{i-1}}\ln(\frac{r_{i-1/2}}{r_{i-1}})}$$

5-12-2 Tangential Transmissibility

If the sector angle in Figure 5.2 is $\Delta\theta$, the upscaled tangential mobility is:

(5-56) $$\lambda^t_{j+1/2} = \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}}$$

(5-57) $$\lambda^t_{j-1/2} = \frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}}$$

5-12-3 Theta Mobility

For the upscaled value $\lambda^\theta_{i+1/2}$, $\lambda^\theta_{j+\frac{1}{2}}$ in the third and second term of equation (5-35), respectively, the harmonic average similar to equation (5-55) should be used, i.e.

(5-58) $$\lambda^\theta_{i+1/2} = \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda^\theta_i}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda^\theta_{i+1}}\ln(\frac{r_{i+1}}{r_{i+1/2}})}$$

(5-59) $$\lambda^\theta_{i-1/2} = \frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda^\theta_i}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda^\theta_{i-1}}\ln(\frac{r_{i-1/2}}{r_{i-1}})}$$

The second term arithmetic average will be:

(5-60) $$\lambda^{\theta}_{j+1/2} = \frac{2 \times \lambda^{\theta}_{j} \lambda^{\theta}_{j+1}}{\lambda^{\theta}_{j} + \lambda^{\theta}_{j+1}}$$

(5-61) $$\lambda^{\theta}_{j-1/2} = \frac{2 \times \lambda^{\theta}_{j} \lambda^{\theta}_{j-1}}{\lambda^{\theta}_{j} + \lambda^{\theta}_{j-1}}$$

5-13 Face Velocity Equation in Polar Coordinate System

The face velocities between block (i, j) and the neighbouring block is calculaeted according to following equation series:

5-13-1 Descritzed Face Velocity Equations in r Direction

(5-62) $$u'_{i+1/2,j} = \lambda^{r}_{i+1/2,j} \frac{(p_{i+1,j} - p_{i,j})}{r_{i+1} - r_{i}} + \frac{1}{2r_{i+1/2}} \left[\lambda^{\theta}_{i,j+1/2} \frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta} + \lambda^{\theta}_{i,j-1/2} \frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta} \right]$$

(5-63) $$u'_{i-1/2,j} = \lambda^{r}_{i-1/2,j} \frac{(p_{i,j} - p_{i-1,j})}{r_{i} - r_{i-1}} + \frac{1}{2r_{i-1/2}} \left[\lambda^{\theta}_{i,j+1/2} \frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta} + \lambda^{\theta}_{i,j-1/2} \frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta} \right]$$

5-13-2 Face Velocity Formula in r Direction in Isotropic Case

For an isotropic medium:

$K_{\theta} = 0$

Therefore,

$\lambda^{\theta} = 0$

Thus equations (5-62) and (5-63) are converted to:

$$(5\text{-}64) \qquad u^{r}_{i+1/2,j} = \lambda^{r}_{i+1/2,j} \frac{(p_{i+1,j} - p_{i,j})}{r_{i+1} - r_i}$$

$$(5\text{-}65) \qquad u^{r}_{i-1/2,j} = \lambda^{r}_{i-1/2,j} \frac{(p_{i,j} - p_{i-1,j})}{r_i - r_{i-1}}$$

In equations (5-64) and (5-65) λ^{r} can be substituted from equations (5-54) and (5-55):

$$(5\text{-}66) \quad u^{r}_{i+1/2,j} = \lambda^{r}_{i+1/2,j} \frac{(p_{i+1,j} - p_{i,j})}{r_{i+1} - r_i}$$

$$\lambda^{r}_{i+1/2} = \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda^{r}_{i}} \ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda^{r}_{i+1}} \ln(\frac{r_{i+1}}{r_{i+1/2}})}$$

$\Rightarrow$

$$u^{r}_{i+1/2,j} = \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda^{r}_{i}} \ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda^{r}_{i+1}} \ln(\frac{r_{i+1}}{r_{i+1/2}})} \frac{(p_{i+1,j} - p_{i,j})}{r_{i+1} - r_i} =$$

$\Rightarrow$

$$\left(\frac{1}{(r_{i+1} - r_i)} \times \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda^{r}_{i}} \ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda^{r}_{i+1}} \ln(\frac{r_{i+1}}{r_{i+1/2}})}\right) \times (p_{i+1,j}) - \left(\frac{1}{(r_{i+1} - r_i)} \times \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda^{r}_{i}} \ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda^{r}_{i+1}} \ln(\frac{r_{i+1}}{r_{i+1/2}})}\right) \times (p_{i,j}) = u^{r}_{i+1/2,j}$$

When $r_{i+1/2}$ is multiplied by $u^{r}{}_{i+1/2,j}$, the result will be:

$$\left[-\left(\frac{r_{i+1/2}}{(r_{i+1}-r_i)}\times\frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right)\times(p_{i,j})\right]$$

(5-67)

$$+\left(\frac{r_{i+1/2}}{(r_{i+1}-r_i)}\times\frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right)\times(p_{i+1,j})=r_{i+1/2}u^{r}{}_{i+1/2,j}$$

Now in this step, $u^{r}{}_{i-1/2,j}$ formula will be found;

For $u^{r}{}_{i-1/2,j}$:

$$u^{r}{}_{i-1/2,j}=\lambda_{i-1/2,j}^{r}\frac{(p_{i,j}-p_{i-1,j})}{r_i-r_{i-1}}$$

$\Rightarrow$

$$\lambda_{i-1/2}^{r}=\frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}$$

$\Rightarrow$

$$u^{r}{}_{i-1/2,j}=\left(\frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right)\times\frac{(p_{i,j}-p_{i-1,j})}{r_i-r_{i-1}}$$

$\Rightarrow$

(5-68) $$\left(\frac{1}{(r_i-r_{i-1})}\times\frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right)\times(p_{i,j})-\left(\frac{1}{(r_i-r_{i-1})}\times\frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right)\times(p_{i-1,j})=u^{r}{}_{i-1/2,j}$$

When $r_{i-1/2}$ is multiplied by $u'_{i-1/2,j}$, it will be:

(5-69)
$$\left[\left(\frac{r_{i-1/2}}{(r_i - r_{i-1})} \times \frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right)\right] \times (p_{i,j})$$
$$-\left(\frac{r_{i-1/2}}{(r_i - r_{i-1})} \times \frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right) \times (p_{i-1,j}) = r_{i-1/2}u'_{i-1/2,j}$$

5-13-3 Descritized Face Velocity Equation in θ Direction

(5-70)
$$u^{\theta}_{i,j+1/2} = \lambda^{\theta}_{i+1/2,j}\frac{(p_{i+1,j} - p_{i,j})}{2(r_{i+1} - r_i)} + \lambda^{\theta}_{i-1/2,j}\frac{(p_{i,j} - p_{i-1,j})}{2(r_i - r_{i-1})} + \frac{1}{r_i}\lambda^t_{i,j+1/2}\frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta}$$

(5-71)
$$u^{\theta}_{i,j-1/2} = \lambda^{\theta}_{i+1/2,j}\frac{(p_{i+1,j} - p_{i,j})}{2(r_{i+1} - r_i)} + \lambda^{\theta}_{i-1/2,j}\frac{(p_{i,j} - p_{i-1,j})}{2(r_i - r_{i-1})} + \frac{1}{r_i}\lambda^t_{i,j-1/2}\frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta}$$

5-13-4 Face Velocity Formula in θ Direction in Isotropic Case

For an isotropic medium:

$K_{\theta} = 0$

Therefore

λ^{θ}=0

Thus equations (5-70) and (5-71) will be reduced to:

(5-72) $$u^{\theta}{}_{i,j+1/2} = \frac{1}{r_i} \lambda^t_{i,j+1/2} \frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta}$$

(5-73) $$u^{\theta}{}_{i,j-1/2} = \frac{1}{r_i} \lambda^t_{i,j-1/2} \frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta}$$

$\lambda^t_{j+1/2}$ could be substituted from equations (5-56) and (5-57), thus for u^{θ}:

$$u^{\theta}{}_{i,j+1/2} = \frac{1}{r_i} \lambda^t_{i,j+1/2} \frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta}$$

$$\lambda^t_{j+1/2} = \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}}$$

$\Rightarrow$

(5-74) $$u^{\theta}{}_{i,j+1/2} = \frac{1}{r_i} \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}} \frac{(p_{i,j+1} - p_{i,j})}{\Delta\theta} = (\frac{1}{\Delta\theta} \frac{1}{r_i} \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}})(p_{i,j+1}) - (\frac{1}{\Delta\theta} \frac{1}{r_i} \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}})(p_{i,j})$$

When $\frac{1}{r_i} \frac{1}{\Delta\theta}$ is multiplied by $u^{\theta}{}_{i,j+1/2}$, it results in:

(5-75) $$\left[-(\frac{1}{\Delta\theta^2} \frac{1}{r_i^2} \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}}) \right] \times (p_{i,j}) + (\frac{1}{\Delta\theta^2} \frac{1}{r_i^2} \frac{\lambda^t_j \lambda^t_{j+1}}{\lambda^t_j + \lambda^t_{j+1}})(p_{i,j+1}) = \frac{1}{r_i} \frac{1}{\Delta\theta} u^{\theta}{}_{i,j+1/2}$$

Now $u^{\theta}{}_{i,j-1/2}$, can be found.

From equation (5-57), $\lambda^t_{j-1/2}$ can be substituted in equation (5-76).

Thus for u^{θ},

(5-76) $$u^{\theta}{}_{i,j-1/2} = \frac{1}{r_i} \lambda^t_{i,j-1/2} \frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta}$$

$$\lambda^t_{j-1/2} = \frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}}$$

$$\Rightarrow$$

$$u^{\theta}_{i,j-1/2} = \frac{1}{r_i}\frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}}\frac{(p_{i,j} - p_{i,j-1})}{\Delta\theta} = (\frac{1}{\Delta\theta}\frac{1}{r_i}\frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}})(p_{i,j}) - (\frac{1}{\Delta\theta}\frac{1}{r_i}\frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}})(p_{i,j-1})$$

When $\frac{1}{r_i}\frac{1}{\Delta\theta}$ is multiplied by $u^{\theta}_{i,j-1/2}$, it will be:

$$\text{(5-77)} \quad \left[(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}})\right](p_{i,j}) - (\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda^t_j \lambda^t_{j-1}}{\lambda^t_j + \lambda^t_{j-1}})(p_{i,j-1}) = \frac{1}{r_i}\frac{1}{\Delta\theta}u^{\theta}_{i,j-1/2}$$

5-14 Discritization of The General Lapacian Equation

According to equation (5-29), the discritization of the general Laplacian can be written as follows;

$$\text{(5-78)} \quad \frac{1}{r_i(r_{i+1/2} - r_{i-1/2})}\left[r_{i+1/2}u^r_{i+1/2,j} - r_{i-1/2}u^r_{i-1/2,j}\right] + \frac{1}{r_i}\frac{1}{\Delta\theta}(u^{\theta}_{i,j+1/2} - u^{\theta}_{i,j-1/2}) = 0$$

Therefore, in equation (5-78) all the previous known factors can be substituted according to their specific equations as previously explained in this chapter. Consequently, face velocity equation will be found according to the equations (5-67) and (5-69):

$$r_{i+1/2}u^{r}{}_{i+1/2,j} - r_{i-1/2}u^{r}{}_{i-1/2,j} =$$

$$\left[-\left(\frac{r_{i+1/2}}{(r_{i+1}-r_i)} \times \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right) - \left(\frac{r_{i-1/2}}{(r_i-r_{i-1})} \times \frac{\ln(\frac{r_{i-1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i-1/2}}{r_i}) + \frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1}}{r_{i-1/2}})}\right) \right] \times (p_{i,j})$$

$$+\left(\frac{r_{i+1/2}}{(r_{i+1}-r_i)} \times \frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right) \times (p_{i+1,j})$$

$$+\left(\frac{r_{i-1/2}}{(r_i-r_{i-1})} \times \frac{\ln(\frac{r_{i-1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i-1/2}}{r_i}) + \frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1}}{r_{i-1/2}})}\right) \times (p_{i-1,j}) = r_{i+1/2}u^{r}{}_{i+1/2,j} - r_{i-1/2}u^{r}{}_{i-1/2,j}$$

(5-79)

For equations (5-75) and (5-77):

$$\frac{1}{r_i}\frac{1}{\Delta\theta}(u^{\theta}{}_{i,j+1/2} - u^{\theta}{}_{i,j-1/2}) =$$

$\Rightarrow$

$$\left[-\left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}\right) - \left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right) \right] \times (p_{i,j}) + \left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right)(p_{i,j-1})$$

$$+\left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}\right)(p_{i,j+1}) =$$

$\Rightarrow$

(5-80) $$\frac{1}{r_i}\frac{1}{\Delta\theta}(u^{\theta}{}_{i,j+1/2} - u^{\theta}{}_{i,j-1/2}) = \frac{1}{r_i^2}\frac{1}{\Delta\theta^2}\left[\left(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}\right)(p_{i,j+1}) - \left(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t} + \frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right)(p_{i,j}) + \frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}(p_{i,j-1}) \right]$$

Substituting equations (5-79) and (5-80) into equation (5-78):

(5-78)
$$\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left[r_{i+1/2}u^r{}_{i+1/2,j}-r_{i-1/2}u^r{}_{i-1/2,j}\right]+\frac{1}{r_i}\frac{1}{\Delta\theta}(u^\theta{}_{i,j+1/2}-u^\theta{}_{i,j-1/2})=0$$

$\Rightarrow$

$$\left[\begin{array}{l} -\dfrac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left(\dfrac{r_{i+1/2}}{(r_{i+1}-r_i)}\times\dfrac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right. \\ -\dfrac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left(\dfrac{r_{i-1/2}}{(r_i-r_{i-1})}\times\dfrac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right) \\ -\dfrac{1}{\Delta\theta^2}\dfrac{1}{r_i^2}\left(\dfrac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}+\dfrac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right) \end{array}\right]\times(p_{i,j})$$

$$+\left[\left(\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\right)\left(\frac{r_{i+1/2}}{(r_{i+1}-r_i)}\times\frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right)\right]\times(p_{i+1,j})$$

$$+\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}\left(\frac{r_{i-1/2}}{(r_i-r_{i-1})}\times\frac{\ln(\frac{r_i}{r_{i-1}})}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right)\times(p_{i-1,j})+$$

$$+\left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right)(p_{i,j-1})+\left(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}\right)(p_{i,j+1})=0$$

(5-81)

Equation (5-82) can be written similar to equation (5-50) for a neighboring grid block:

$$(5\text{-}50) \quad r_{i+1/2} = \frac{(r_{i+1} - r_i)}{\ln(\frac{r_{i+1}}{r_i})}$$

$$(5\text{-}82) \quad r_{i-1/2} = \frac{(r_i - r_{i-1})}{\ln(\frac{r_i}{r_{i-1}})}$$

Substituting equations (5-50) and (5-82) in equation (5-81) results in:

$$\left[\begin{array}{l} -\frac{1}{r_i(r_{i+1/2} - r_{i-1/2})}\left(\frac{r_{i+1/2}}{\frac{(r_{i+1} - r_i)}{\ln(\frac{r_{i+1}}{r_i})}} \times \frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right. \\ -\frac{1}{r_i(r_{i+1/2} - r_{i-1/2})}\left(\frac{r_{i-1/2}}{\frac{(r_i - r_{i-1})}{\ln(\frac{r_i}{r_{i-1}})}} \times \frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right) \\ -\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\left(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t + \lambda_{j+1}^t} + \frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t + \lambda_{j-1}^t}\right) \end{array}\right] \times (p_{i,j})$$

$$+\left[\begin{array}{l} \left(\frac{1}{r_i(r_{i+1/2} - r_{i-1/2})}\left(\frac{r_{i+1/2}}{\frac{(r_{i+1} - r_i)}{\ln(\frac{r_{i+1}}{r_i})}} \times \frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right)\right) \\ +\left(\frac{1}{2r_i^2\Delta\theta}\left(\frac{1}{(r_{i+1} - r_i)} - \frac{1}{(r_i - r_{i-1})}\right)\left(\frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^\theta}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{i+1}^\theta}\ln(\frac{r_{i+1}}{r_{i+1/2}})}\right)\right) \end{array}\right] \times (p_{i+1,j})$$

$$+\frac{1}{r_i(r_{i+1/2}-r_{i-1/2})}(\frac{r_{i-1/2}}{\frac{(r_i-r_{i-1})}{\ln(\frac{r_i}{r_{i-1}})}}\times\frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})})\times(p_{i-1,j})+$$

$$+(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t})(p_{i,j-1})+(\frac{1}{\Delta\theta^2}\frac{1}{r_i^2}\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t})(p_{i,j+1})=0 \tag{5-83}$$

$\Rightarrow$

Multipling by $\Delta\theta^2 r_i^2$:

$$\begin{bmatrix} -\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})} \\ -\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}) \\ -(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}+\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}) \end{bmatrix}\times(p_{i,j})$$

$$+\begin{bmatrix} (\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})}) \\ +(\frac{\Delta\theta}{2}(\frac{1}{(r_{i+1}-r_i)}-\frac{1}{(r_{i+1}-r_i)})(\frac{\ln(\frac{r_{i+1}}{r_i})}{\frac{1}{\lambda_i^\theta}\ln(\frac{r_{i+1/2}}{r_i})+\frac{1}{\lambda_{i+1}^\theta}\ln(\frac{r_{i+1}}{r_{i+1/2}})})) \end{bmatrix}\times(p_{i+1,j})$$

$$\begin{aligned} &+\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{1}{\frac{1}{\lambda_i^r}\ln(\frac{r_i}{r_{i-1/2}})+\frac{1}{\lambda_{i-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})})\times(p_{i-1,j})+ \\ &+(\frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t})(p_{i,j-1})+(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t})(p_{i,j+1})=0 \end{aligned} \tag{5-84}$$

The above equation (5-84) can be written as:

(5-85) $a(i,j)p_{i,j} + b(i,j)p_{i,j+1} + c(i,j)p_{i,j-1} + d(i,j)p_{i-1,j} + e(i,j)p_{i+1,j} = 0$

For $i = 2,....,N; j = 1,....,M$

For the first ring, close to the inner boundary condition and near the well bore, equation (5-85) becomes:

(5-86) $a(1,j)p_{1,j} + b(1,j)p_{1,j+1} + c(1,j)p_{1,j-1} + e(1,j)p_{2,j} = -d(1,j)p_w$

For ; $j = 1,....,M$

For the last ring, close to the outer boundary condition:

(5-87) $a(N,j)p_{N,j} + b(N,j)p_{N,j+1} + c(N,j)p_{N,j-1} + d(N,j)p_{N-1,j} = -e(N,j)p_e$

For ; $j = 1,....,M$

where M is the number of sectors and N is the number of the rings.

The boundary condition on the inner boundary is $P_{0,j} = P_w$ and on the outer boundary is $P_{N+1,j} = P_e$.

According to equation (5-85) all coefficients a, b, c, d and e can be found as:

(5-88) $$a(i,j) = -\frac{\Delta\theta^2 r_i}{(r_{i+1/2} - r_{i-1/2})}\left(\frac{1}{\frac{1}{\lambda_j^r}\ln(\frac{r_{i+1/2}}{r_i}) + \frac{1}{\lambda_{j+1}^r}\ln(\frac{r_{i+1}}{r_{i+1/2}})} + \frac{1}{\frac{1}{\lambda_j^r}\ln(\frac{r_i}{r_{i-1/2}}) + \frac{1}{\lambda_{j-1}^r}\ln(\frac{r_{i-1/2}}{r_{i-1}})}\right) - \left(\frac{\lambda_j^t\lambda_{j+1}^t}{\lambda_j^t + \lambda_{j+1}^t} + \frac{\lambda_j^t\lambda_{j-1}^t}{\lambda_j^t + \lambda_{j-1}^t}\right)$$

(5-89) $b(i,j) = (\dfrac{\lambda_j^t \lambda_{j+1}^t}{\lambda_j^t + \lambda_{j+1}^t})$

(5-90) $c(i,j) = (\dfrac{\lambda_j^t \lambda_{j-1}^t}{\lambda_j^t + \lambda_{j-1}^t})$

(5-91) $d(i,j) = \dfrac{\Delta\theta^2 r_i}{(r_{i+1/2} - r_{i-1/2})} \times \dfrac{1}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_i}{r_{i-1/2}}) + \dfrac{1}{\lambda_{i-1}^r}\ln(\dfrac{r_{i-1/2}}{r_{i-1}})}$

(5-92) $e(i,j) = (\dfrac{\Delta\theta^2 r_i}{(r_{i+1/2} - r_{i-1/2})} \times \dfrac{1}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_{i+1/2}}{r_i}) + \dfrac{1}{\lambda_{i+1}^r}\ln(\dfrac{r_{i+1}}{r_{i+1/2}})})$

The equations series (5-88) to (5-92) can be simplified using $\lambda_{i+1/2}^r$ and $\lambda_{i-1/2}^r$ as:

(5-93) $\lambda_{i+1/2}^r = \dfrac{\ln(\dfrac{r_{i+1}}{r_i})}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_{i+1/2}}{r_i}) + \dfrac{1}{\lambda_{i+1}^r}\ln(\dfrac{r_{i+1}}{r_{i+1/2}})} \Rightarrow \dfrac{\lambda_{i+1/2}^r}{\ln(\dfrac{r_{i+1}}{r_i})} = \dfrac{1}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_{i+1/2}}{r_i}) + \dfrac{1}{\lambda_{i+1}^r}\ln(\dfrac{r_{i+1}}{r_{i+1/2}})}$

(5-94) $\lambda_{i-1/2}^r = \dfrac{\ln(\dfrac{r_i}{r_{i-1}})}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_i}{r_{i-1/2}}) + \dfrac{1}{\lambda_{i-1}^r}\ln(\dfrac{r_{i-1/2}}{r_{i-1}})} \Rightarrow \dfrac{\lambda_{i-1/2}^r}{\ln(\dfrac{r_i}{r_{i-1}})} = \dfrac{1}{\dfrac{1}{\lambda_i^r}\ln(\dfrac{r_i}{r_{i-1/2}}) + \dfrac{1}{\lambda_{i-1}^r}\ln(\dfrac{r_{i-1/2}}{r_{i-1}})}$

Finally equation series (5-88) to (5-92) will convert to:

(5-95) $a(i,j) = -\dfrac{\Delta\theta^2 r_i}{(r_{i+1/2} - r_{i-1/2})}(\dfrac{\lambda_{i+1/2}^r}{\ln(\dfrac{r_{i+1}}{r_i})} + \dfrac{\lambda_{i-1/2}^r}{\ln(\dfrac{r_i}{r_{i-1}})}) - (\dfrac{\lambda_j^t \lambda_{j+1}^t}{\lambda_j^t + \lambda_{j+1}^t} + \dfrac{\lambda_j^t \lambda_{j-1}^t}{\lambda_j^t + \lambda_{j-1}^t})$

$$(5\text{-}96)\quad b(i,j)=\left(\frac{\lambda_j^t \lambda_{j+1}^t}{\lambda_j^t+\lambda_{j+1}^t}\right)$$

$$(5\text{-}97)\quad c(i,j)=\left(\frac{\lambda_j^t \lambda_{j-1}^t}{\lambda_j^t+\lambda_{j-1}^t}\right)$$

$$(5\text{-}98)\quad d(i,j)=\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{\lambda_{i-1/2}^r}{\ln\left(\frac{r_i}{r_{i-1}}\right)}$$

$$(5\text{-}99)\quad e(i,j)=\left(\frac{\Delta\theta^2 r_i}{(r_{i+1/2}-r_{i-1/2})}\times\frac{\lambda_{i+1/2}^r}{\ln\left(\frac{r_{i+1}}{r_i}\right)}\right)$$

Note that $r_0=r_w$ and $r_{n+1}=r_e$.

In the θ direction $p_{i,M+1}=p_{i,1}$ and $p_{i,0}=p_{i,M}$.

Consequently, according to equation (5-84) and knowing equation series (5-95) to (5-99) as coefficient of $p_{i,j}$, the unknown vector could be found in different ways using the MATLAB code.

After solving the Laplacian and finding the pressure for each grid block in a Polar grid according to the previous equation series and writing the appropriate MATLAB code for this step, velocities for each grid block in both directions r and θ will be found.

According to equation series (5-62) and (5-63) for r direction, and equation series (5-70) and (5-71) for θ direction, the appropriate pressure can be substituted for the assumed grid block in these equations and the relevant velocity for that cell can be found.

The velocity distribution for all finite difference grid cells in both r and θ directions can now be from the pressure distribution.

The next step is to determine streamline time of travel for both directions in one grid cell. Time of travel in r direction is found according to equation (5-42). Parameters a and b are found from the velocity equations, (equations (5-39), (5-40) and (5-41)). Similarly for θ direction, time of travel is found according to equation (5-43) using velocity equation in θ direction and finding a and b according to equation (5-37) and (5-38).

Consequently, when time of travel for each grid cell in r and θ direction are found, then the time of flight for the streamlines' particle is fixed according to equation (5-44).

After finding time of flight for the streamline at each cell, next exit location, r_{exit} and θ_{exit}, can be found according to equations (5-45) and (5-46). This calculation process shall be repeated step by step for the next grid cell and at the end the streamlines' route can be visualized in Polar coordinates. Figure 3.7 shows the calculation sequence in a flow chart.

Chapter VI

Case Studies

6-1 Summary of Previous Chapters

As mentioned in previous chapters, this research includes streamline simulation in Cartesian and Polar coordinates. The solution for each coordinate system comprises of two main parts: Theoretical part and MATLAB code programming.

The goal of the theoretical part is to develop a new method for generating and analyzing streamlines in reservoirs near well bores using a finite difference method and the Pollock method in each coordinate system.

In Cartesian coordinate system, as it is mentioned in chapter III and chapter IV, the pressure distribution is found by solving the Laplace equation using finite difference method, (details of the pressure solution are described in chapter IV). Then, one streamline particle is assumed at the entrance point of one grid block at the outer boundaries. Exit coordinate of streamline's particle for the grid block can be found according to the Pollock method, equation series (3-6) to (3-20). Hence, the new exit point will be the entrance point for the new grid block. This calculation procedure can be reiterated for next grids.

This procedure continues until the next boundary condition (e.g. the well bore) is encountered. At the final step, all these entrance and exit points connect together and constitute the streamline.

The same procedure can be used for new entrance points at the boundary to produce more streamline paths. By using this method streamline behavior and fluid flow, all across the reservoir can be simulated in a Cartesian coordinate system.

In a Polar coordinate system, the relevant equations for pressure distribution are developed for isotropic medium case according to chapter V. Then, equations for velocity and time of travel in r and θ directions are developed, (equation series (5-35) to (5-46)). In the third step, radial, tangential and theta transmissibility equations are developed and substituted in Darcy's equation and face velocity equations, (equation series (5-47) to (5-77)). Consequently, all these known factors are substituted in general discritized Laplacian equation, equation (5-78) and equation (5-84).

After solving the Laplacian and finding the pressure for every grid block in a Polar coordinate system, the velocities for each grid block in both r and θ directions are found, (equation series (5-62) to (5-71)). The pressure for the assumed grid block, which has been calculated before, can be substituted in these equations and the velocity for that cell can be found.

At the next step, time of travel in r direction, (the time needed for a particle to travel with the velocity $u_r(r)$ from the entrance point to the opposite angular face) and time of travel in θ direction, (the time needed for the particle to travel with velocity $u_\theta(\theta)$ from the entrance point to the opposite face in radial direction) are calculated. Consequently time of flight, (the minimum of time of travel in r and θ directions) for streamlines in every cell is found from equation series (5-38) to (5-42). Subsequently next exit location can be found via these equations.

For other grid cells the same calculations could be repeated step by step and at the end, the streamlines' path can be visualized in the Polar coordinate system according to the specified boundary conditions.

The second main part of this research is programming and developing streamline simulation code using MATLAB. In the theoretical part of the research, all relevant equations for modeling are developed. In this part, the MATLAB code is developed according to all theoretical equations. Some examples are shown in this chapter and details of the MATLAB code are presented in the Appendix. The developed

MATLAB code has strong ability to visualize all streamlines with different boundary conditions and heterogeneous reservoirs.

Two separate types of MATLAB codes are developed for streamline simulation in Cartesian and Polar coordinate systems. The code developed in the Cartesian coordinate system includes two main parts: main file and subroutines. Permeability, pressure and velocity for each node are found by subroutines and other calculations are conducted in the main file.

The MATLAB code developed in the Polar coordinate system includes main route and two subroutines. All the calculations are done in main route and the sub routines do plotting.

Various outputs are resulted from the codes depending on the case studies and requirements. The main output is a visualization of streamlines for different boundary conditions.

Five case studies are presented in this thesis. The numerical results are compared with theoretical expectation.

6-2 Case Studies in Cartesian Geometries

6-2-1 Homogenous and Isotropic Medium

In this case study (Figure 6.1), the inflow boundary is the vertical boundary at the left. A no-flow boundary condition is assumed at both horizontal boundaries. The inflow boundary has higher pressure than the exit boundary on the right side. This case study is assumed to be in a homogenous medium.

Basic Assumptions

1- Isotropic and homogenous medium for the given reservoir

2- No-flow boundary condition for horizontal boundaries

Permeability Distribution

$$K_x(i,j) = 0.2\times10^{-14} m^2$$
$$K_y(i,j) = 0.2\times10^{-14} m^2$$

(i, j) : Grid cells numbering

Grid Cells Numbering (N, M)

The problem is solved for two cases of 100 by 100 ($M=N=100$) and 50 by 50 ($M=N=50$) grid cells.

N is the number of grid cell(s) in horizontal direction.

M is the number of grid cell(s) in vertical direction.

Grid Cell Length (L)

Dimensions of each grid cell in x and y directions are one meter.

$\Delta x = 1.0 \ m$

$\Delta y = 1.0 \ m$

The relevant total length of reservoir is equal to the number of grid cells in each direction multiplied by the grid cell length:

$$\begin{array}{l} L_x = 100 \times 1m = 100m \\ L_y = 100 \times 1m = 100m \end{array} \Rightarrow \ 100m \times 100m$$

or

$$\begin{array}{l} L_x = 50 \times 1m = 50m \\ L_y = 50 \times 1m = 50m \end{array} \Rightarrow \ 50m \times 50m$$

Boundary Conditions

Pressure

At the entry points of the inflow boundary:

$P(i,1) = 1 \ bar$

and on the exit boundary:

$P(i,N) = 0.01 \ bar$

Viscosity

$\mu = 0.0005 \ Pa.s$

Porosity

$\phi = 0.2$

Physical Expectation

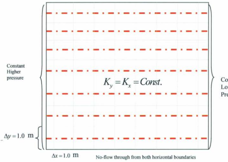


Figure 6.1 : Theoretical Expectation for First Case Study in Cartesian Coordinate System

For numerical calculations, in isotropic medium, pressures for every grid cells are found according to equations (4-11) to (4-19) and are substituted in equation (4-20). According to equation (4-20), the differential pressure has direct effect on velocity. Since, streamlines' time of travel is a function of velocity, and then from equation (3-20), the exit location can be found. Therefore, it is expected to see smooth and straight streamlines with no curvature or vertical component of dicplacement.

MATLAB Program Result

For details of MATLAB code, refer to Appendix.

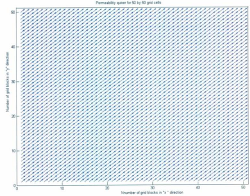


Figure 6.2: Permeability Quiver for 50 by 50 Grid Cells for First Case Study in Cartesian Coordinate System

As it is shown in Figure 6.2, K_x and K_y, permeabilities in x and y directions have the same values and the quiver of permeability is the same for all grid cells.

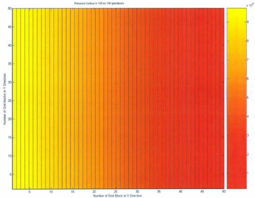


Figure 6.3: Pressure Contour in 50 by 50 Meshes for First Case Study in Cartesian coordinate system

Figure 6.3, shows pressure decreases linearly in the x -direction and is constant in y – direction.

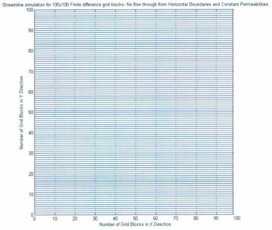


Figure 6.4: Streamline Behavior in 100 by 100 Gird Blocks for First Case Study in Cartesian coordinate system

Figure 6.4 shows streamline behavior in 100 by 100 grids according to the previous assumptions. As shown in the Figure 6.4, all streamlines are horizontal from high pressure to low pressure boundary condition with no vertical movements. There is one streamline per vertical grid cell.

6-2-2 Low Permeability Region in the Middle of an otherwise homogeneous Reservoir

In this case study the streamline behaviour is analyzed in an isotropic, heterogeneous reservoir. A low permeability region is located in the central part of reservoir and the background permeability has a higher value.

Basic Assumption

1- No-flow through horizontal boundaries

2- Heterogeneous and Isotropic medium

Permeability Distribution

In this case study different permeabilities are assumed in different locations.

Low Permeability Region

$$K_x(i,j) = 0.1\times10^{-20} m^2$$
$$K_y(i,j) = 0.1\times10^{-20} m^2$$

Background Permeability

$$K_x(i,j) = 0.2\times10^{-14} m^2$$
$$K_y(i,j) = 0.2\times10^{-14} m^2$$

(i,j) : Grid cells numbering

Grid Cells Numbering (N, M)

The problem is solved for two cases of 100 by 100 (M=N=100) or 50 by 50 (M=N=50) grid cells.

N is the number of grid cell(s) in horizontal direction.

M is the number of grid cell(s) in vertical direction.

Grid Cell Length (L)

Horizontal and vertical dimensions of each grid cell are one meter.

$\Delta x = 1.0 \ m$

$\Delta y = 1.0 \ m$

Total Length

The relevant total length of reservoir is equal to:

$$\begin{aligned} L_x &= 100 \times 1m = 100m \\ L_y &= 100 \times 1m = 100m \end{aligned} \quad \Rightarrow \quad 100m \times 100m$$

or

$$\begin{aligned} L_x &= 50 \times 1m = 50m \\ L_y &= 50 \times 1m = 50m \end{aligned} \quad \Rightarrow \quad 50m \times 50m$$

Boundary Condition

At the entry points of the inflow boundary:

$P(i,1) = 1 \ bar$

and on the exit boundary:

$P(i, N) = 0.01\ bar$

N is the number of grid cell(s) in horizontal rows.

Viscosity

$\mu = 0.0005\ Pa.s$

Porosity

$\phi = 0.2$

Physical Expectation

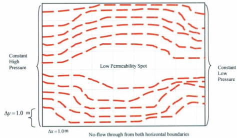


Figure 6.5: Theoretical Expectation for Second Case Study in Cartesian Coordinate System

A constant high pressure in the inflow boundary on the left side of the reservoir and a constant lower pressure on the outflow boundary are assumed. A low permeability region is assumed at the middle of the reservoir, and other parts have constant higher permeability. The Pressure distribution is calculated according to equations (4-10), (4-11) and (4-18). In one grid cell streamlines exit point is found according to equation (4-6). When there is lower permeability, lower velocities will be found. In the equation (3-16), with low velocity, a larger time of travel will be found. In other words, reaching to the next grid cell that has lower permeability takes more time in the low permeability region. Hence, according to equation (3-22), minimum time of travel is the actual time of flight, which means that streamlines' particle prefers to pass through higher permeability region and move around low permeability regions. Therefore, according to the given boundary conditions, it is expected to see streamlines bypass the low permeability region, (Figure 6.5).

MATLAB Program Result

For details of MATLAB code, refer to Appendix.

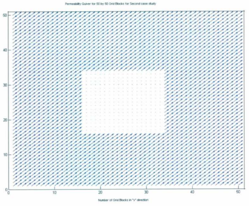


Figure 6.6: Permeability Quiver for 50 by 50 Grid Cells for Second Case Study in Cartesian Coordinate System

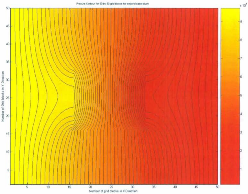


Figure 6.7: Pressure Contour for 50 by 50 Grid Cells for Second Case Study in Cartesian coordinate system

As it is shown in the Figure 6.7, pressure contours in the low permeability area are closer which is because of higher-pressure gradient when permeability is lower.

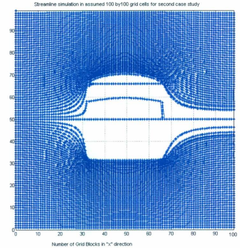


Figure 6.8: Streamline Simulation in 100 by100 Grid Cells for Second Case Study in Cartesian coordinate system

Figure 6.8, shows the ouput of numerical simulation where streamlines tend to move around the low permeability region and by pass it. There is a minor asymmetry in the streamlines in the middle of the reservoir which is suppoed to be in low permeability area. The assymmetry comes from a numerical error in MATLAB script that could not be tracked

6-2-3 Assuming a Well in the Corner of the Region

In this case study, (Figure 6.9), the inflow boundary is the vertical boundary on the left hand side. In the exit boundary, on the right side of the reservoir, a well is assumed with lower pressures at the upper corner and the rest of the right hand side vertical boundary is assumed as no-flow boundary. The no-flow boundary condition is assumed at both horizontal boundaries. This case study is assumed to be in an isotropic, homogeneous reservoir.

Basic Assumption

1- Isotropic and homogeneous medium

Permeability Distribution

$$K_x(i,j) = 0.2\times10^{-14}m^2$$
$$K_y(i,j) = 0.2\times10^{-14}m^2$$

(i,j) : Grid cells numbering

Grid Cells Numbering (N, M)

The problem is solved for two cases of 100 by 100 (M=N=100) or 50 by 50 (M=N=50) grid cells.

N is the number of grid cell(s) in horizontal direction.

M is the number of grid cell(s) in vertical direction.

Grid Cell Length

Horizontal and vertical dimensions of each grid cell are equal to one meter.

$\Delta x = 1.0\ m,\ \Delta y = 1.0\ m$

Total Length

The relevant total length of reservoir is:

$$\begin{matrix} L_x = 100 \times 1m = 100m \\ L_y = 100 \times 1m = 100m \end{matrix} \Rightarrow 100m \times 100m$$

or

$$\begin{matrix} L_x = 50 \times 1m = 50m \\ L_y = 50 \times 1m = 50m \end{matrix} \Rightarrow 50m \times 50m$$

Boundary Condition

At the inflow boundary:

$P(i,1) = 1\ bar$

and on the exit boundary:

$P(T, N) = 0.001\ bar$

T is the number of grid cells in vertical rows in well bore region.

$\frac{7M}{10} \le T \le M$ (M is the total number of grid cells in vertical rows)

Viscositiy

$\mu = 0.0005\ Pa.s$

Porosity

$\phi = 0.2$

Physical Expectation

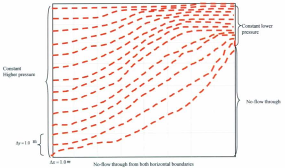


Figure 6.9: Theoretical Expectation in Third Case Study in Cartesian Coordinate System

According to equation (3- 6), larger differential pressures result in larger velocities. Then, based on equation (3-16), time of travel in the direction with larger differential pressure is less than the other direction and consequently streamlines' particle tend to go in the direction with larger differential pressure.

Therefore, according to the boundary conditions and assumptions, it is expected to see streamlines moving toward the upper, right corner and collect in the sink, (well) area.

MATLAB Program Result

For details of MATLAB code, refer to the Appendix.

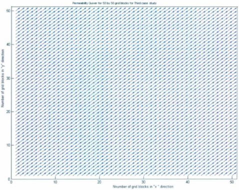


Figure 6.10: Permeability Quiver for 50 by 50 Grid Cells for Third Case Study in Cartesian Coordinate System

As it is shown in the figure 6.10, K_x and K_y in both directions have same values and the quiver of both K_x and K_y has the same direction in all grid cells.

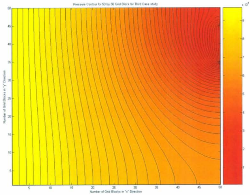


Figure 6.11: Pressure Contour for Assumed 50 by 50 Grid Cells in Cartesian Coordinate System

Figure 6.11 shows pressure decreasing from high-pressure boundary at the left to lower pressure boundary at the well area.

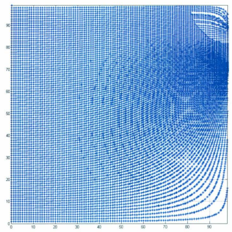


Figure 6.12: Streamline Simulation for 100 by 100 Grid Cells for Third Case Study in Cartesian coordinate system

As it is shown in Figure 6.12, streamlines smoothly tend to go toward the well area and collect in the lower pressure boundary (sink).

6-3 Case Studies in Radial Geometries

6-3-1 Isotropic and Homogeneous Medium

In this case study, (Figure 6.13), higher pressure is assumed at the outer boundary (inflow) and lower pressure is assumed in the center (well). Isotropic, homogeneous medium is assumed for the given reservoir.

Basic Assumption

1- Isotropic homogeneous medium

2- Constant pressure at the boundaries

Permeability Distribution

In this case, permeability is constant.

$K_r(i, j) = 2\times10^{-12} m^2$

Boundary Condition

Pressure

Pressure at the center of the ring, (well);

$P_w = 0.2 \quad bar$

Pressure at the inflow boundary;

$P_e = 1.0 \quad bar$

P_w : Well pressure

P_e : Pressure at the outer boundary

Radius

$r_w = 0.2 \ m$

$r_e = 1.10 \ m$

Viscosity

$\mu = 0.001 \ Pa.s$

Porosity

$\phi = 0.2$

Grid Cells Numbering

Number of the rings (N) =10

Number of sectors (M) = 60

Grid Cells Measurements

The angle for each sector is $\frac{2\pi}{60}$

The radius, r for the rings increases exponentially

Physical Expectation

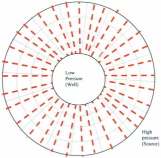


Figure 6.13: Theoretical Expectation for the First Case Study in Radial Geometry

The pressure distribution is found according to equation series (5-88) to (5-91). Then, these found pressures are substituted in equation series (5-64), (5-65), (5-72) and (5-73) and velocities in r and θ-directions are found. Radial and tangential transmissibility are found according to equation series (5-54) to (5-57). Note that theta transmissibility is not used in this case because of isotropic case. After finding time of flight according to equations (5-42), (5-43) and (5-44), the results are substituted in equation (5-45) and (5-46) and r and θ for exit coordinates are found. With high pressure at the outer boundary and lower pressure in the center of the rings, it is expected to see streamlines being radial, gathering in the low-pressure area (well).

MATLAB Program Result

For details of MATLAB code, refer to the Appendix.

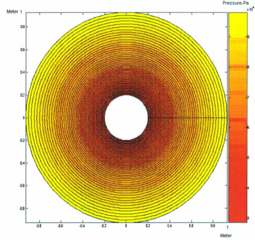


Figure 6.14: Pressure Contour in Polar Coordinates System for the First Case Study in Radial Geometry

Pressure distribution is shown in Figure 6.14, decreasing from high pressure at the outer boundary to lower pressure in the center.

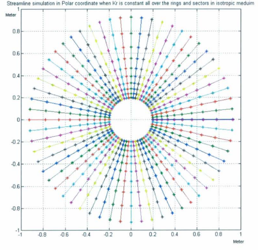


Figure 6.15: Streamline Simulation in Polar Coordinate for the First Case Study in Radial Geometry System

As it is shown in Figure 6.15, all streamlines move toward the center of the rings radially.

6-3-2 Low Permeability Region in the Middle of an otherwise homogeneous, isotropic Reservoir

In this case study the streamline behaviour is analyzed in an isotropic, heterogenous reservoir in Polar coordinate system. A low permeability region is located in the reservoir and the background permeability has a higher value.

Basic Assumption

1- Heterogeneous, Isotropic medium

2- Constant pressure at the boundaries

Permeability Distribution

1- Permeability in low permeability region;

$K_r(i,j) = 0.1\times10^{-16}\,m^2$

2- Permeability in other areas:

$K_r(i,j) = 2\times10^{-12}\,m^2$

(i,j) : Grid Cells numbering in Polar coordinate system in r and θ direction

Boundary Conditions

Pressure

Pressure at the center of the ring, (well);

$P_w = 0.2 \quad bar$

Pressure at the inflow boundary;

$P_e = 1.0 \quad bar$

P_w: Well pressure

P_e: Pressure at the outer boundary

Radius

$r_w = 0.2 \quad m$

$r_e = 100.0 \quad m$

Viscosity

$\mu = 0.001 \quad Pa.s$

Porosity

$\phi = 0.2$

Grid Cells Numbering

Number of the rings (N) =80

Number of sectors (M) = 80

Grid Cells Measurements

The angle for each sector is $\frac{2\pi}{80}$

The radius, r for different rings increases exponentially.

Physical Expectation

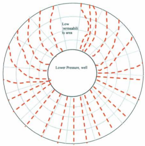


Figure 6.16: Theoretical Expectation for the Second Case Study in Radial Geometry

When there is low permeability region in the grid cells, then radial and tangential transmissibility have smaller values according to equations (5-54) to (5-57). Therefore velocities in r direction have smaller values compared to θ direction, based on equation series (5-64), (5-65), (5-72) and (5-73). Then, it takes more time for streamlines' particle (time of travel) to reach to the other side of the grid cell and according to equation (5-44) time of flight will be equal to the time of travel in θ direction. Consequently, streamlines' particle changes its path to in the θ direction and gets around the lower permeability region.

MATLAB Program Result

For details of MATLAB code, refer to the Appendix.

Pressure distribution is shown in Figure 6.17. Generally, pressure contours are decreasing from high pressure at the outer boundary to lower pressure in the center. As it is presented in the graph, heterogeneity causes deformed contour shapes because of higher pressure gradient at low permeability area.

When low permeability region is assumed in the middle of the reservoir, as is shown in Figure 6.18, streamlines try to move around the low permeability region. In figure 6.18 the main goal has been capturing the path of streamlines near the low permeability region.

There is a minor asymmetry in the streamlines on the other side of the reservoir that comes from a numerical error in MATLAB script that could not be tracked. The conclusion then is that there is a minor programming error which adds noise on top of outherwise sound solutions. The behavior shown in the figures would not have been possible had there been a major modeling error.

As mentioned before the shape of streamlines in the low permeability region conforms to what we expect from theory.

Low permeability area

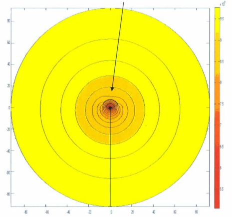


Figure 6.17: Pressure contour in Polar Coordinate for the second Case Study in Radial Geometry

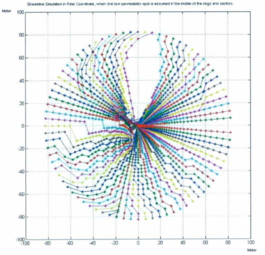


Figure 6.18: Streamline Simulation in Polar Coordinate for Second Case Study in Radial Geometry

Chapter VII

Research Novelty, Conclusions and Recommendations

7-1 Summary and Conclusions

This research presents a methodology in Cartesian and Polar coordinates for generating and analyzing streamlines in reservoirs near wellbores in finite difference grids using the Pollock method.

Given an entry point of a streamline into a grid cell, Pollock's method starts by mapping the grid cell onto the unit square. Each component of the velocity field is then approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point. The exit point is mapped back into physical space to give the entry point of the next cell, and so on.

This method demonstrates how streamlines behave when mechanical and geological factors change in the reservoir and effects on route of stream path lines. By using this proposed method, stream path lines can be visualized for any given boundary conditions.

This method has the strong ability for considering large finite difference grids with thousands of grid blocks and various potential fluid flow mechanical factors, such as pressure, velocity and viscosity and also various geological factors.

All these processes are working for both Cartesian and Polar geometries with two different variables (x, y) and (r, θ), respectively.

In this research, five case studies have been examined to ensure acceptable and stable results when applied to real physical examples.

7-2 Research Novelty

To our knowledge, streamline simulation in near well bore regions using radial geometry has not been done before.

Numerical analytical tools are regularly used in the oil and gas industry as means to establish the optimal procedure to maximize the efficiency of the project in computation time, less error calculation and in a cost effective manner. Several researches show streamline-based simulation is more powerful and time-efficient than older modeling methods. Some of the benefits of streamline method in calculation process are as follows (Thiele 2001):

- **Flow Visualization:**

Flow visualization is one of the interesting issues for academic and industries because it shows flow path lines and the geometric condition of wells. Using flow visualization allows for the opportunity to follow streamline behaviour in a reservoir and gives the ability to analyze well conditions and reservoir modeling with respect to streamline movements.

- **Full Field Modeling**

When there is a problem in a reservoir, most of the time, considering a small model in a limited area to solve a problem in that location will not give real results because of all interaction and effects that come from outer boundaries; thus, the best way to solve the problem is modeling the entire reservoir. When all parts of the reservoir are considered in calculation, all the influences and cell communications will also be considered.

- **Efficiency and Computational Speed**

Streamline simulation method has more ability to simplify computation and is more efficient and faster in approaching the result compared with more conventional calculation methods.

- **Flow Physics- Starting With the Simplest Model**

In streamline simulation, the problem, solving can start with the simplest model, which is the fastest. Then progressing flow physics and model complexities can be added.

7-3 Recommendations for Future Research

This research developed a comprehensive approach to simulate streamlines near a well bore in a reservoir. With further research, the proposed method has great potential to improve upon the initial assumptions, which will lead to more accurate and practical results.

7-3-1 In this research, oil streamline is simulated in two dimensions in Cartesian coordinate in x and y direction. This research can be modified in three dimensions, adding z dimension to model streamline behaviour in reservoirs.

7-3-2 In this research, isotropic medium is assumed. The next step could be refined by choosing an anisotropic medium for the reservoir.

7-3-3 In this research, fluid is assumed to be incompressible; therefore, further research that uses compressible flows in a case study close to real conditions is recommended.

References

Arihara, N. (2005). Reservoir Simulation Technology by Streamline-based Methods, J. Jpn Pet Inst, 48; NO.6; Page 325-335

Ask, A. K., Dahle, H. K., Karlsen K. H., Nordhaug H. F. (2000). A Local Streamline Eulerian-Lagrangian Method for Two-Phase Flow- Dept. of Mathematics, University of Bergen, Norway- To appear in Proc. of XIII Int. Conf. Computational Methods in Water Resources

Aavatsmark, I. (2002). An introduction to multipoint flux approximations for quadrilateral grids. Comput. Geosci., 6:405–432

Aziz, K., Settari, A. (1997). Petroleum Reservoir Simulation. Calgary, ISBN: 0853347875 : 9780853347873

Baker, R. (2001). Streamline R Technology: Reservoir History Matching and forecasting Its Success, Limitation, and Future, SPE Journal, Volume 40, No.4

Batycky, R. P., Thiele, M. R., Blunt, M. J. (1997). A Streamline-Based Reservoir Simulation of the House Mountain Waterflood, SCRF- Stanford University center for Reservoir Forecasting (SCRF) Annual Report,

Batycky, R. P. (1997). A three-dimensional two-phase field scale streamline simulator. PhD thesis, Stanford University, Dept. of Petroleum Engineering

Bensabat, J., Zhou, Q., Bear, J. (2000). An adaptive pathline-based particle-tracking algorithm for the Eulerian-lagrangian method. Adv. Water Resource, 23:383–397

Berre, I., Dahle, Karlsen, H. K., Lie, K. A., Natvig, J. R. (2002). Time-of-Flight + Fast Marching+ Transport Collapse: An Alternative to Streamlines for Two-Phase Porous- Media Flow with Capillary Forces

Bratvedt, F., Gimse, T., Tegnander, C. (1996). Transport in Porous Media; Streamline computations for porous media flow including gravity v25, No.1 p 63-78

Bratvedt, F., Bratvedt, K., Buchholz, C.F., Gimse, T., Holden, H. L. and Risebro, N. H. (1993). Frontline and Frontsim two full scale, two- phase, black oil reservoir simulators based on front tracking. Surv. Math. Ind. 3:185-215

Cheng, H. P., Cheng, J. R., Yeh, G. T. (1996). A Particle tracking technique for the Lagrangian-Eulerian Finite Element Method in Multi-Dimensions- International Journal for Numerical Methods in Engineering Volume 39, Issue 7, pages 1115–1136

Cordes, C., Kinzelbach, W. (1992). Continous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. Water Resource. Res., 28(11):2903–2911

Dahle, H. K., Espedal, M. S., Ewing, R. E., Sævereid , O. (1990). Characteristic adaptive subdomain methods for reservoir flow problems; Numerical Methods for Partial Differential Equations Volume 6 Issue 4, Pages 279 – 309

Dahle, H. K., Magne, S., Espedal, M. S., Sævareid , O. (1992). Characteristic, local grid refinement techniques for reservoir flow problems-International Journal for Numerical Methods in Engineering, Volume 34 Issue 3, Pages 1051 – 1069

Datta-Gupta, A., King, M. J. (1995). A semi analytic approach to tracer flow modeling in heterogeneous permeable media. Advances in Water Resources, Volume 18, Issue 1, Pages 9-24

Das, B. M. (1997). Advanced Soil Mechanics, Second Edition, Washington, DC : Taylor & Francis ISBN: 1560325615 9781560325611,

Dawson, C. N. (1991). Godunov-mixed methods for advective flow problems in one space dimension. SIAM J. Num Anal.28(5)

Douglas, J, Furtado, F., Pereira, F. (1997). Computational Geosciences On the numerical simulation of water flooding of heterogeneous petroleum reservoirs-Computational geosciences 1, 19-48

Durlofsky, L. J. (1994). Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities. Water Resource. Res., 30(4):965–973

Edwards, M. G. (2002). Unstructured, control-volume distributed full-tensor finite-volume schemes with flow based grids. Computer, Geosciences, 6:433–452

Espedal, M.S., Karlsen, K. H. (1998) - Numerical solution of reservoir flow models based on large time step operator splitting algorithms- Filtration in Porous Media and Industrial Applications -Cetraro, Italy

Fayprats, H. (1951). The application of numerical methods to cycling and flooding problems-- 3rd World Petroleum Congress,

Goode, D. J. (1990). Particle velocity interpolation in block-centered finite difference groundwater flow model- Water Resource Research 26, no. 5: 925-940

Hewett, T. A.,Yamada, T. (1997). Theory for the semi-analytical calculation of oil recovery and effective relative permeability's using streamtubes- Advances in Water resources- Volume 20, Number 5

Higgins, R. V., Leighton, A. J. (1962). A Computer Method to Calculate Two-Phase Flow in Any Irregularly Bounded Porous Medium- Journal of Petroleum Technology

James, G., Brnet, L., McBryan, O. A., Plohr, Bradley, Yaniv, S. (1983). Front Tracking for Petroleum Reservoir Simulation. SPE Reservoir Simulation Symposium, 15-18 November, SanFransico, California

Johansen, T. E. (2008). Principal of Reservoir Engineering; Course Note, Memorial University of Newfoundland

Johansen, T. E. (2009). Research Note; Memorial University of Newfoundland

Kaasschieter, E.F. (1995). Mixed finite elements for accurate particle tracking in saturated groundwater flow. Advanced Water Resource, 18(5):277–294

King, M. J., Datta-Gupta, A. (1998). Streamline simulation: A current perspective. In Situ, 22(1):91–140

Kipfer, P., Reck, F., Greiner, G. (2003). Local exact particle tracing on unstructured grids Comput. Graph. Forum, 22(2):1–9

Knight, D., Mallinson, G. (1996.) Visualizing unstructured flow data using dual stream functions. IEEE Trans. Vis. Comput. Graph. 2(4):355–363

Lu, N. (1994). A semi analytical method of path line computation for transient finite difference ground water flow models - Water Resources Research, Vol. 30, NO. 8, PP. 2449-2459, 1994

Mallison, B. T. (2006). Improved Mappings for Streamline-Based Simulation, SPE Journal, Volume 11, Number 3, pp. 294-302,. Society of Petroleum Engineers

Mohseni, K., Colonius, T. (2000). Numerical treatment of Polar coordinate singularities, Journal of Computational Physics, Volume 157 , Issue 2- table of contents Pages: 787 - 795 Year of Publication: ISSN:0021-9991

Moreno, J., Kazemi, h., Giman, J. R. (2004). Streamline Simulation of Countercurrent Water-Oil and Gas-Oil Flow in Naturally Fractured Dual- Porosity Reservoirs. SPE Annual Technical Conference and Exhibition, Huston, Texas, USA

Muscat, M., Wyckoff R. D. (1934). AIME Technical Publication

Naff, R. L., Russell, T. F., Wilson, J. D. (2002.), Shape functions for velocity interpolation in general hexahedral cells. Comput. Geosci., 6(3-4):285–314

Oliveira, A., Baptista, A. M. (1998.). On the role of tracking on eulerian-lagrangian solutions of the transport equation. Adv. Water Resource, 21:539–554

Pollock, D. W. (1998). Semi analytical Computation of Path Lines for Finite Difference Models; Ground Water journal, Vol.26, No.6

Russell, T. F., Healy, R. W. (2000). Analytical tracking along streamlines in temporally linear Raviart-Thomas velocity fields

Sadarjoen, I. A., vanWalsum, T. A., Hin, J. S., Post, F. H. (1997). Particle tracing algorithms for 3d curvilinear grids. Scientific visualization: Overviews, methodologies, and techniques, chapter 14, pages 311–335. IEEE

Schafer, A. L., Perini, Wilson, J. L. (1991). Efficient and accurate front tracking for two dimensional groundwater flow models. Water Resource, Research, 27(7):1471–1485,

Shekhar, C. P., Kumar, O., N. (1997), Modeling of Free Surface in Steady-State Radial Seepage, Journal of Hydrologic Engineering, Vol. 2, No. 1, pp. 39-41, (doi 10.1061/(ASCE)1084-0699(1997)2:1(39))

Shafer, J. M. (1987). Reverse path line calculation of time-related capture zones in non uniform flow. Ground Water, 25(3):283–289,

Shirayama, S. (1993). Processing of computed vector fields for visualization. J. Comput. Phys., 106:30–41

Strid, T., Rizzi, A., Oppelstrup, J. (1989). Development and use of some flow visualization algorithms. Lecture series/Von Karman Institute for Fluid Dynamics, (7):1–56

Thiele, M. R. (2001). Streamline Simulation; 6th International Forum on Reservoir Simulation- Schloss Fuschi, Austria

Thiele, 1994; Batycky, (1997)-ResAssist website (http://www.res-assist.com/))

Verma, S., Aziz, K. (1997). A control-volume scheme for flexible grids in reservoir simulation, SPE 37999. In SPE Reservoir Simulation Symposium, Dallas, Texas

Zheng, C., Bennett, G. D. (2002). Applied contaminant transport modeling, Wiley Interscience, New York, second edition

Appendix

Source Code for Streamline Simulation

This appendix includes five parts, which is described in following pages.

This MATLAB code is developed originally by MARJAN HASHEM.

```
% Streamline Simulation Near Well Bore
%      By MARJAN HASHEM

% Developed MATLAB Program for Streamline
Simulation
% This Code developed originally by MARJAN
HASHEM

% First Case Study in Cartesian Coordinate
% Main route
clc;
clear all;
close all;
format long e

Y = input('Number of Nodes in Y ');
X = input('Number of Nodes in X ');

% Boundary values can be set along the
corner or in the middle
reply = input('Corner or Middle
boundary conditions C/M [C]: ', 's');

%Maximum Error abs(x2-x1)
maxerr= input('Desired Maximum Error
[.00001]: ');
if isempty(reply)
    reply = 'C';
end
if isempty(maxerr)
    maxerr = .00001;
end

global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
%  fid2 = fopen('ST OUTPUT.dat','w');
%  fprintf(fid2,'  Xe        Ye        Ax
Vxp     Ay      Vyp /L\n\n');

permeabilities

reply=lower(reply);
if reply=='c'
pressure
else
    for i=floor(X/3):floor(2*X/3)
    P(i,1)=(2*i);
end
end


M=(0.5*10^-3);
velocities

for b=1:X+1
    x(b)=(b-1);
end
for t=1:Y+1
    y(t)=(t-1);
end


Xp(1,1)=0;
Yp(1,1)= 0;
  Xpp=Xp(1,1);
  Ypp=Yp(1,1) ;
 Vxp(1,1)=Vx(1,1);
 Vyp(1,1)=Vy(1,1) ;
 i=1;
 j=1;

X;
Y;

  while ((Xpp< (X-1) ) & (Ypp<Y) &
(i<(Y+1)) & (j<(X-3)))

        location
if (rr==1);
    break
  end
111;
  if (Ye(i,j)>=0 & Xe(i,j)>=0)

      gg=(Xe(i,j)-x(j));

                 if  (gg>=0.98 && gg<=1.1)
                   777;
                   Xp(i,j+1)=Xe(i,j);
                   Yp(i,j+1)=Ye(i,j);
                   Xpp=Xp(i,j+1);
                   Ypp=Yp(i,j+1);
                     i=i;
                   j=j+1;
                 else
                   5555;
                   Xp(i+1,j)=Xe(i,j);
                   Yp(i+1,j)=Ye(i,j) ;
                   Xpp=Xp(i+1,j);
                   Ypp=Yp(i+1,j);
                   i=i+1;
                   j=j;

                 end
                 rr==0;
  else
      8787;
    rr==1;
     break

end

end


fprintf(fid2,'  Xe        Ye         Ax
Vxp     Ay      Vyp /L\n\n');
fprintf(fid2,'%f  %f  %e    %e   %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
  plotFinal
  hold on
  Xp=zeros(X,Y);
  Yp=zeros(X,Y);

    Vxp=zeros(X,Y);
  Vyp=zeros(X,Y);


  for m=2:Y ;
```

```
Xp(m,1)=0;
Yp(m,1)=(m-1);
 Xpp=Xp(m,1);
 Ypp=Yp(m,1) ;
Vxp(m,1)=Vx(m,1);
Vyp(m,1)=Vy(m,1) ;
 i=m;

j=1;

    while ((Xpp<(X-1) ) & (Ypp<Y) &
(i<(Y+1)) & (j<(X-1)))

 location ;
% sign

if rr==1;

    break
end

if (Ye(i,j)>=0 & Xe(i,j)>=0)

    gg=(Xe(i,j)-x(j));

            if  (gg>=0.98 && gg<=1.1)
             777;
             Xp(i,j+1)=Xe(i,j);
             Yp(i,j+1)=Ye(i,j);
             Xpp=Xp(i,j+1);
             Ypp=Yp(i,j+1);
              i=i;
             j=j+1;
            else
              5555;
              Xp(i+1,j)=Xe(i,j);
              Yp(i+1,j)=Ye(i,j) ;
              Xpp=Xp(i+1,j);
              Ypp=Yp(i+1,j);
              i=i+1;
              j=j;

            end
            rr==0;

else

   rr==1;
     break

end

          x(j);
       end
       fprintf(fid2,' Xe      Ye        Ax
Vxp     Ay     Vyp /L\n\n');
       fprintf(fid2,'%f  %f  %e   %e  %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
           plotFinalforYdirection
           hold on
           Xp=zeros(X,Y);
           Yp=zeros(X,Y);
 end

      for m=Y+1 ;

 Xp(m,1)=0;
 Yp(m,1)=(m-1);
  Xpp=Xp(m,1);
  Ypp=Yp(m,1) ;
 Vxp(m,1)=Vx(m-1,1);
 Vyp(m,1)=Vy(m,1) ;
  i=m;
%  i=14
 j=1;

%        while ((Xpp<=(X-1) ) & (Ypp<=Y)
& (i<=(Y+1)) & (j<(X)))
             while (Xpp<(X-1) )

 location ;
% sign
if (rr==1);
    break
end
if (Ye(i,j)>=0 & Xe(i,j)>=0)

    gg=(Xe(i,j)-x(j));

            if  (gg>=0.98 & gg<=1.1)
             777;
             Xp(i,j+1)=Xe(i,j);
             Yp(i,j+1)=Ye(i,j);
             Xpp=Xp(i,j+1);
             Ypp=Yp(i,j+1);
              i=i;
             j=j+1;
            else
              5555;
              Xp(i+1,j)=Xe(i,j);
              Yp(i+1,j)=Ye(i,j) ;
              Xpp=Xp(i+1,j);
              Ypp=Yp(i+1,j);
              i=i+1;
              j=j;

            end
            rr==0;
else

    rr==1;
     break

end

          x(j);

if (rr==1);
   break
end
    end
    fprintf(fid2,' Xe      Ye        Ax
Vxp     Ay     Vyp /L\n\n');
       fprintf(fid2,'%f  %f  %e   %e  %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
           plotFinalforYdirection
end
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for finding Permeability

```
function permeabilities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp

    for i=1:Y
    for j=1:X
        Kx(i,j)=(0.2*(10^-2));
         Ky(i,j)=(0.2*(10^-2));
    end
    end
    Kx
    Ky
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for finding the Velocity

```
function velocities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp


M=(0.5*10^-3);
  Vx=zeros(Y+1,X+1);
  Vy=zeros(Y+2,X);
   for i=1:(Y+1)
        Vx(i,1)=(0.005) ;

end
for i=2:Y+1
     for j=2:X
   Vx(i-1,j)=(-1)*((Kx(i-1,j-1)*(P(i-1,j)-P(i-1,j-1)))/(M*x));
     end
end
for i=Y+1
     for j=1:(X+1)
         Vx(i,j)=Vx(i-1,j);
     end
end

 for i=2:Y
           for j=2:X+1
Vy(i,j-1)=(-1)*((Ky(i-1,j-1)*(P(i,j-1)-P(i-1,j-1)))/(M*y));


      end
 end

Vx
 Vy
quiver (Vx,Vy)
 hold on
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for finding the pressure distribution all across the grid blocks

```
function pressure
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
    P=zeros(Y,X);
for i=1:Y
        P(i,1)=100000;
          for i=1:Y
              P(i,X)= 1000;
              end

end


      maxr=1;
```

```
errormatrix=zeros(X,Y);
itteration=0;
while maxr>maxerr
    for i=1:Y
        for j=2:X-1
            tempval=P(i,j);
            x=1.;
            y=1.;


           if i==1
            P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i+1,j))/4);

           elseif i==Y
            P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i-1,j))/4);


           else

P(i,j)=((P(i,j+1)+P(i,j-
1)+P(i+1,j)+P(i-1,j))/4);
           end



        end
    end
    maxr=max(max(errormatrix));
    itteration=itteration+1;
end



for i=1:Y
        for j=1:X
            P(i,j);
        end
    end

       contourf (P,50, 'DisplayName',
'PRESSURE');colormap autumn;
       figure
itteration;
 P



% Streamline Simulation Near Well Bore
%     By MARJAN HASHEM

% Developed MATLAB Program for Streamline
Simulation
% This Code developed originally by MARJAN
HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for finding the exit coordinate of the
streamline Particle

function location
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
% fid2 = fopen('any name.dat','w');

% input('first  model')
 disp ('first model')
 9876
if  (Vx(i,j)> 0 & Vx(i,j+1)> 0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)> 0 &
Vy(i+1,j)> 0 & Vy(i,j)==Vy(i+1,j))
      disp(' 1-1 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
       if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end

       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


 elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
       disp(' 2-1 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
```

```
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
        disp(' 3-1 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 );
        disp('-4-1')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j);
         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);


    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
         disp('-5-1')
```

```
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%       - second model')
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
        disp('-6-1')
%         if Vyp(i,j)<0
%                           -6-1-1')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
%        Xe(i,j)= x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
  if Vyp(i,j)<0
                          disp('-6-1-1')
        Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);

                          else Vyp(i,j)>0
```

```
                        disp('-6-1-2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
%         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
%         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
%         Xe(i,j)= x(j+1);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
          disp('-7-1')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
          disp('-8-1')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          disp('-9-1')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
          disp('-10-1')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-1')


                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          disp('-12-1')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
          disp('-13-1')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



         disp(' - second model')

%         - first model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
          disp('-1-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp(' -2-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
%          Tm(i,j)=min (Tx(i,j),Ty(i,j))
         Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
%          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
          disp('-5-2')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
%       - second model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp(' -6-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
            Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
if Vyp(i,j)<0
                          disp('-6-2-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                  else Vyp(i,j)>0
                       disp('-6-2-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
%         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
     disp('-7-2')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
          disp('-8-2')

          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i,j))
          if Ty(i,j)<0
            Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
          else
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          end

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          disp('-9-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
           if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
         Xe(i,j);
         Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
          disp(' -10-2')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);


%          - third model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
           disp('-11-2')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
%          Tm(i,j)=min (Tx(i,j),Ty(i,j));
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          disp(' -12-2')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
          disp(' -13-2')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);
```

```
        disp(' - THIRD model')

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
        disp(' -1-3')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
        disp(' -2-3 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
         disp('-3-3')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
         disp('-4-3')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
         disp(' -5-3')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
   Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
            disp('-6-3')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%        Xe(i,j)= x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
 if Vyp(i,j)<0
                         disp('-6-3-1')
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                   else Vyp(i,j)>0
                         disp('-6-3-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
       disp(' -7-3')

                       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
         end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
        disp('-8-3 ')

                       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
         end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
       disp(' -9-3')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          disp('-10-3')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-3')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          disp('-12-3')
                          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          disp('-13-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        disp(' - FORTH model')

%         - first model
elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
            disp(' -1-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
          disp('-2-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          disp('-3-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
```

```
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
         disp(' -4-4 ')
          disp('I dont know')
          444
           rr==1
           555
           return
           666



           elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          disp('-5-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
         disp('-6-4')

         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Xe(i,j)=x(j);
if Vyp(i,j)<0
                         disp('-6-4-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                     else Vyp(i,j)>0
                        disp(' -6-4-2')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
```

```
        disp(' -7-4')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

        Tm(i,j)=Ty(i,j)
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
         disp('-8-4')

       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

        Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
         disp('-9-4')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);
       Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);

          elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
         disp('-10-4')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

        Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);



%         - third model
       elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
         disp('-11-4')
         3333

        Xe(i,j)=x(j+1);
222
 rr==1


       elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
        disp(' -12-4')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
        disp(' -13-4')
       disp ('I dont know')
  rr==1



 disp('fifth model')

elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
        disp(' -1-5')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
```

```
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
%        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        disp(' -2-5')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
      %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)>0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
        disp(' -3-5')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
         disp('-4-5')
         disp('I dont know')
       rr==1

          elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         disp('-5-5')
          disp('I dont know')
   rr==1



%       - second model
       elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
         disp('-6-5')

       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
      Tm(i,j)=Tx(i,j)
      %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
```

```
Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%        Xe(i,j)= x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
if Vyp(i,j)<0
                              disp(' -6-5-1')
        Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);

                           else Vyp(i,j)>0
                              disp(' -6-5-2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
      Tm(i,j)=Tx(i,j)
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i+1);
       Xe(i,j);
       Ye(i,j);
         end
          elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
        disp(' -7-5')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
      %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
        disp(' -8-5 ')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
      %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
        disp(' -9-5')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
```

```
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

          elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-5')
                        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=min (Tx(i,j),Ty(i,j))
          %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);



%       - third model
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-5')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
          end
%          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
          Tm(i,j)=min (Tx(i,j),Ty(i,j));
          %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);


          elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
          disp(' -12-5')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));
          %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
          elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
            disp('-13-5 ')
            disp('I dont know')


rr==1
%



 disp('-sixth  model')


elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
          disp('-1-6')
```

```
            if ( Vxp(i,j)<0)
               disp(' -1-6-1')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);
          Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
          Xe(i,j);
          Ye(i,j);
             else ( Vxp(i,j)> 0)
                  disp('-1-6-2')
             Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);
          Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
          Xe(i,j);
          Ye(i,j);
             end
    elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
           disp('-2-6')
             if ( Vxp(i,j)<0)
                  disp('-2-6-1')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             else ( Vxp(i,j)> 0)
              disp('-2-6-2')
               Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             end
       elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
            disp('-3-6')
             if ( Vxp(i,j)<0)
                 disp('-3-6-1')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
            if ( Vxp(i,j)<0)
                disp(' -3-6-1')
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             else ( Vxp(i,j)> 0)
                 disp('-3-6-2')
              Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             end
          elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
            disp('-4-6')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
          if ( Vxp(i,j)<0)
          disp(' -4-6-1')
         Xe(i,j)=x(j);
        Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);
       else( Vxp(i,j)>0)
            disp('-4-6-2')
            Xe(i,j)=x(j+1);
        Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);
       end


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          disp('-5-6')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
             Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
              if ( Vxp(i,j)<0)
           disp('-5-6-1')
%         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
% %         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
       else( Vxp(i,j)>0)
           disp('-5-6-2')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
% %         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

       Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
       end

%
%       - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
          disp('-6-6')
          if Vyp(i,j)<0
              disp(' -6-6-1')
               Ye(i,j)=y(i);
          else  Vyp(i,j)>0
               disp('-6-6-2')
              Ye(i,j)=y(i+1);
          end
          if Vxp(i,j)<0
              disp(' -6-6-3')
               Xe(i,j)=x(j);
          else  Vxp(i,j)>0
              disp(' -6-6-4')
              Xe(i,j)=x(j+1);
          end

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
         disp(' -7-6 ')
          if  Vxp(i,j)<0
              disp(' -7-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          else Vxp(i,j)>0
               disp('-7-6-2')
             Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
          end

        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
         disp(' -8-6')
```

```
            if  Vxp(i,j)<0
                 disp('-8-6-1')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
            else Vxp(i,j)>0
                 disp('-8-6-2')
               Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j) ;
            end


          elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
           disp(' -9-6 ')
            if   Vxp(i,j)<0
                disp('-9-6-1')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
           Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
            if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);
          Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
          Xe(i,j);
          Ye(i,j);
            else Vxp(i,j)>0
               disp(' -9-6-2')
              Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);
          Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
          Xe(i,j);
          Ye(i,j)  ;
            end


             elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
           disp('-10-6')


Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            if   Vxp(i,j)<0
                disp('-10-6-1')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
            else Vxp(i,j)>0
                disp('-10-6-2')

          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
            end
```

```
%        - third model
       elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
        disp(' -11-6')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          if  Vxp(i,j)<0
           disp(' -11-6-1')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
        else  Vxp(i,j)>0
           disp(' -11-6-2')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j) ;
         end

       elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
        disp(' -12-6')

               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         if  Vxp(i,j)<0
              disp('-12-6-1')
% %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
%        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j)
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
          else  Vxp(i,j)>0
              disp('-12-6-2')
               Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
          end


        elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         disp('-13-6')


              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          if  Vxp(i,j)<0
         disp('-13-6-1 ')
       Xe(i,j)=x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
     else  Vxp(i,j)>0
         disp('-13-6-2 ')

       Xe(i,j)=x(j+1);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
     end



     disp(' -seventh model')

% - first model
```

```
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp('-1-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp('-2-7 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
         disp('-3-7')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-7 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=Tx(i,j)
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);



           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          disp('-5-7')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
```

```
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-7')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         disp(' -6-7-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                       else Vyp(i,j)>0
                          disp('-6-7-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
```

```
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
         disp(' -7-7 ')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-7 ')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         disp('-9-7')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
       Xe(i,j)
       Ye(i,j)

          elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
        disp(' -10-7 ')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);



%        - third model
       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
        disp(' -11-7')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%    Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
        disp(' -12-7')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
         disp(' -13-7')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


      disp(' - eighth model')

%         - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
          disp('-1-8')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
          disp('-2-8')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
         disp(' -4-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          disp('-5-8 ')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-8')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          if Vyp(i,j)<0
                                disp('-6-8-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                        else Vyp(i,j)>0
                           disp(' -6-8-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
         disp(' -7-8 ')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1)
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-8 ')
```

```
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
        disp('-9-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
        disp('-10-8 ')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);








%          - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
        disp(' -11-8')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
```

```
         disp(' -12-8')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          disp('-13-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

        disp(' -ninth model')
%           - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp(' -1-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp(' -2-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-9')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-9 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          disp('-5-9')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%        - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-9')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         disp('-6-9-1')
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                     else Vyp(i,j)>0
                         disp('-6-9-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j+1)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
          disp('-7-9')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
```

```
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-9 ')

                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         disp(' -9-9')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          disp('-10-9')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-9')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
         disp(' -12-9')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
         disp(' -13-9 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        disp(' -tenth  model')
%        - first model
```

```
elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         disp('-1-10')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        disp(' -2-10')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
            disp('-3-10 ')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
            Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
            end
            Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
            disp('-4-10')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
            Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);

            if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
            end
            Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
            Ye(i,j)=y(i);
            Xe(i,j);
            Ye(i,j);


        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
            disp(' -5-10')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
            Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
            Ty(i,j)=(y(i)-Yp(i,j)/Vy(i,j);
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
            end
            Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);


%       - second model
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
            disp(' -6-10')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
            Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
            if Vyp(i,j)<0
                        disp(' -6-10-1')
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

            if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
            end
            Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%           Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
            Ye(i,j)=y(i);
            Xe(i,j);
            Ye(i,j);

                        else Vyp(i,j)>0
                            disp(' -6-10-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
      Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i+1);
       Xe(i,j);
       Ye(i,j);
         end
          elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
         disp('-7-10 ')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
        disp(' -8-10')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
         disp('-9-10')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);

          elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
         disp('-10-10')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
```

```
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%         - third model
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-10')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
          disp('-12-10')
               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
         disp(' -13-10 ')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


        disp(' -eleventh model')

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         disp(' -1-11')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)==0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         disp(' -2-11')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)==0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          disp('-3-11')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
          disp('-4-11')
          disp('I dont know')
 Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%                 Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         disp(' -5-11')
           disp('I dont know')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
             Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
% %         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
%        - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
         disp(' -6-11')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                        disp(' -6-11-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                     else Vyp(i,j)>0
                        disp(' -6-11-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
          disp('-7-11')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
          disp('-8-11')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
          disp('-9-11')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
```

```
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-11')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);




%         - third model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
         disp(' -11-11 ')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j)
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         disp(' -12-11')
               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
          disp('-13-11')
         disp(' I dont know')

rr==1
%
```

```
 disp('-TWELVE MODEL')
elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
        disp(' -1-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        disp(' -2-12')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)==0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
        disp(' -3-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
         disp('-4-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1))

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


          elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
        disp(' -5-12')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
          disp('-6-12')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                          disp('-6-12-1')
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                      else Vyp(i,j)>0
                         disp(' -6-12-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
         disp(' -7-12')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
         disp(' -8-12')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
          disp('-9-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-12')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-12 ')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         disp(' -12-12')
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(+1j)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
```

```
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
        disp(' -13-12')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        disp(' - thirtheenth  model')



        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)>0
& Vy(i,j)==Vy(i+1,j))
        disp(' -1-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);



    elseif  (Vx(i,j)==0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        disp(' -2-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
    elseif  (Vx(i,j)==0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
        disp('-3-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
        disp('-4-13')
        disp('Idont know')
    rr==1

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
        disp('-5-13')
        disp(' I dont know')

    rr==1



%       - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
        disp('-6-13 ')


        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        if Vyp(i,j)<0
                    disp('-6-13-1')
        Xe(i,j)=x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                else Vyp(i,j)>0
                    disp('-6-13-2')
        Xe(i,j)=x(j);
```

```
            Ye(i,j)=y(i+1);
            Xe(i,j);
            Ye(i,j);
            end
          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
& Vy(i,j)> Vy(i+1,j) )
            disp('-7-13')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);
            elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
& Vy(i,j)< Vy(i+1,j) )
             disp(' -8-13')

            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
             Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);

          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
& Vy(i,j)== Vy(i+1,j) )
            disp('-9-13')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%             Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
            Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);
            Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
            Xe(i,j);
            Ye(i,j);

          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
            disp('-10-13')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%             Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
            Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);

%         - third model
          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
            disp(' -11-13 ')

            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%             Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
            Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);

          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
            disp(' -12-13')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%             Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
            Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
            end
            Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
            Xe(i,j);
            Ye(i,j);
            elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
            disp('-13-13 ')
```

```
        disp('I dont know')

rr==1
%
        else
            disp('new model')

            Vx(i,j)
             Vx(i,j+1)
              Vy(i,j)
               Vy(i+1,j)
        end
        5755823
fprintf(fid2,' Xe      Ye       Ax
Vxp     Ay      Vyp /L\n\n');
fprintf(fid2,'%f  %f  %e    %e   %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
end
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for drawing the streamline

```
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
% temp = find(Xp~=0);
temp = find(Xp);
xx = Xp(temp)';
yy = Yp(temp)';
  xx = [Xp(1,1); xx];
  yy = [Yp(1,1); yy];
plot(xx,yy,'-')
grid on
axis equal
axis square
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate

% Subroutine for drawing streamline in "x" Direction

```
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
temp = find(Xp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'-')
grid on
axis equal
axis square
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for drawing streamline in "y" Direction

```
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
temp = find(Yp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'-')
grid on
axis equal
axis square
```

% Streamline Simulation Near Well Bore
% By MARJAN HASHEM

% Developed MATLAB Program for Streamline Simulation
% This Code developed originally by MARJAN HASHEM

% First Case Study in Cartesian Coordinate
% Subroutine for drawing the streamline

```
function sign
```

```
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr

if (Ye(i,j)>0 & Xe(i,j)>0)

    gg=(Xe(i,j)-x(j));

            if  (gg>=0.98 && gg<=1.1)
              777
              Xp(i,j+1)=Xe(i,j);
              Yp(i,j+1)=Ye(i,j);
              Xpp=Xp(i,j+1);
              Ypp=Yp(i,j+1);
               i=i
              j=j+1
            else
               5555
              Xp(i+1,j)=Xe(i,j);
              Yp(i+1,j)=Ye(i,j) ;
              Xpp=Xp(i+1,j);
              Ypp=Yp(i+1,j);
              i=i+1
              j=j

            end
            rr==0
else

    Yp(i,j)=Ye(i,j);
    Xp(i,j)=Xe(i,j);
    rr==1;
     return

end
```

Second Case Study in Cartesian Coordinate

```
%   Streamline Simulation Near Well
Bore
%            By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   Main route

clc;
clear all;
close all;
format long e

Y = input('Number of Nodes in Y ');
X = input('Number of Nodes in X ');

% Boundary values can be set along the
corner or in the middle
reply = input('Corner or Middle
boundary conditions C/M [C]: ', 's');

%Maximum Error abs(x2-x1)
maxerr= input('Desired Maximum Error
[.00001]: ');
if isempty(reply)
    reply = 'C';
end
if isempty(maxerr)
    maxerr = .00001;
end

global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B
fprintf(fid2,'  Xe       Ye          Ax
Vxp      Ay      Vyp /L\n\n');

permeabilities

reply=lower(reply);
if reply=='c'
pressure
else
    for i=floor(X/3):floor(2*X/3)
     P(i,1)=(2*i);
end
end

M=(0.5*10^-3);
velocities

for b=1:X+1
     x(b)=(b-1);
end
for t=1:Y+1
     y(t)=(t-1);
end
```

```
Xp=zeros(X,Y);
Yp=zeros(X,Y);


 Vxp=zeros(X,Y);
Vyp=zeros(X,Y);

            for m=2:Y

Xp(m,1)=0;
Yp(m,1)=(m-1);
 Xpp=Xp(m,1);
 Ypp=Yp(m,1) ;
Vxp(m,1)=Vx(m,1);
Vyp(m,1)=Vy(m,1) ;
 i=m;

j=1;

         while ((Xpp<(X-1) ) & (Ypp<Y)
& (i<(Y)) & (j<(X-1)))

 location ;
 fprintf(fid2,'%f  %f  %e   %e  %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
if rr==1;

    break
end

if (Ye(i,j)>=0 & Xe(i,j)>=0)

    Xpp
    Ypp
    Xe(i,j)
    Ye(i,j)
    x(j)
    y(i)
    i
    j
    if (abs(Ye(i,j)-Ypp))>1.25
        disp('benvis')
        if (Ye(i,j)-Ypp)<0
            Ye(i,j)=(fix(Ypp)-1)
        elseif (Ye(i,j)-Ypp)>0
            Ye(i,j)=(fix(Ypp)+1)
        else

        end
    end

        if (abs(Xe(i,j)-Xpp))>1.25
       disp('benvis')
       if (Xe(i,j)-Xpp)<0
           Xe(i,j)=(fix(Xpp)-1)
       elseif (Xe(i,j)-Xpp)>0
           Xe(i,j)=(fix(Xpp)+1)
       else

       end
        end


        if (Xe(i,j)-(fix(Xe(i,j))))==0

       B=Xe(i,j)+1
        elseif ((Xe(i,j)-
(fix(Xe(i,j))))>=0.9 & (Xe(i,j)-
(fix(Xe(i,j))))<1)
            B=(round(Xe(i,j)))+1
        elseif ((Xe(i,j)-
(fix(Xe(i,j))))>0 & (Xe(i,j)-
(fix(Xe(i,j))))<=0.01)
            B=(round(Xe(i,j)))+1
        else
       B=(fix(Xe(i,j)))+1
        end


        if (Ye(i,j)-(fix(Ye(i,j))))==0

            A=Ye(i,j)+1
        elseif ((Ye(i,j)-
(fix(Ye(i,j))))>0.9 & (Ye(i,j)-
(fix(Ye(i,j))))<1)

            A=(round(Ye(i,j)))+1
            elseif ((Ye(i,j)-
(fix(Ye(i,j))))>0 & (Ye(i,j)-
(fix(Ye(i,j))))<=0.01)

            A=(round(Ye(i,j)))+1
        else
         A=(fix(Ye(i,j)))+1
        end
         A

                 Xp(A,B)=Xe(i,j);
                 Yp(A,B)=Ye(i,j);
                 Xpp=Xp(A,B);
                 Ypp=Yp(A,B);
                  i=A;
                 j=B;

                    rr==0;

else

    rr==1;
    break

end


          x(j);

      end
           plotFinalforYdirection
           hold on
           Xp=zeros(X,Y);
           Yp=zeros(X,Y);

       end
```

```
%   Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   M-File- For finding the Permeabilty


function permeabilities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B

    for i=1:Y
    for j=1:X
        Kx(i,j)=(0.2*(10^-2));
          Ky(i,j)=(0.2*(10^-2));
    end
    end

     for i=fix((Y)/3):fix((2*Y)/3)
        for j=fix((X)/3):fix((2*X)/3)
            Kx(i,j)=(0.1*(10^-16));
            Ky(i,j)=(0.1*(10^-16));
        end
     end
     K1=(0.1*(10^-16));
     K2=(0.2*(10^-2));


    Kx;
    Ky;
```

```
%   Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   M-File- For Finding the Velocity

function velocities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B


M=(0.5*10^-3)
  Vx=zeros(Y+1,X+1);
  Vy=zeros(Y+2,X);
    for i=1:(Y+1)

      Vx(i,1)=(0.005)  ;


end
for i=2:Y+1
      for j=2:X
  Vx(i-1,j)=(-1)*((Kx(i-1,j-1)*(P(i-
1,j)-P(i-1,j-1)))/(M*x));

        end
end
for i=Y+1
    for j=1:(X+1)
        Vx(i,j)=Vx(i-1,j);
    end
end




 for i=2:Y
        for j=2:X+1
Vy(i,j-1)=(-1)*((Ky(i-1,j-1)*(P(i,j-1)-
P(i-1,j-1)))/(M*y));



    end
 end

 Vx;
  Vy;
```

```
%       j=1:Y+1


%   Streamline Simulation Near Well
Bore
%         By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   M-File For Finding the Exit
coordinate of the streamlines' Particle



function location
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B
% fid2 = fopen('any name.dat','w');



% input('first  model')
 disp ('first model')

if  (Vx(i,j)> 0 & Vx(i,j+1)> 0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)> 0 &
Vy(i+1,j)> 0 & Vy(i,j)==Vy(i+1,j))
      disp(' 1-1 ');
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
       if Ty(i,j)<0

         Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

          Tx(i,j) =Tx(i,j)*(-1);
       end

       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);



 elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
       disp(' 2-1 ')  ;
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
        disp(' 3-1 ') ;
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
                Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
```

```
        Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 );
          disp('-4-1');
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
          disp('-5-1');
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-1') ;
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);

       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
      Tm(i,j)=Tx(i,j);
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
  if Vyp(i,j)<0
                         disp('-6-1-1');
        Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


                    else Vyp(i,j)>0
                         disp('-6-1-2');
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
      Tm(i,j)=Tx(i,j);
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i+1);
       Xe(i,j);
       Ye(i,j);
         end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
         disp('-7-1') ;
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
```

```
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
          disp('-8-1');

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          disp('-9-1') ;
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
          disp('-10-1') ;
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-1');


                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          disp('-12-1') ;
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
          disp('-13-1') ;

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


         disp(' - second model');
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp('-1-2');
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
             1317
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
        disp(' -2-2') ;
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          % disp('-3-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0

            Ty(i,j) =Ty(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%           Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          % disp('-4-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
          % disp('-5-2')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
         % disp(' -6-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
if Vyp(i,j)<0
                          % disp('-6-2-
1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                     else Vyp(i,j)>0
                          % disp('-6-2-
2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));

        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
```

```
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
          % disp('-7-2')
                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
%               Tm(i,j)=Tx(i,j);
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
        else

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
        end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
          % disp('-8-2')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i,j));
 if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0

               Ty(i,j) =Ty(i,j)*(-1);
         end


         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          % disp('-9-2')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
         % disp(' -10-2')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0

            Ty(i,j) =Ty(i,j)*(-1);
         end
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));
           %         Tm(i,j)=min
(Tx(i,j),Ty(i,j))
%          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
          % disp('-11-2')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
          if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
 if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
 end

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
```

```
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
         % disp(' -12-2')
                         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
         % disp(' -13-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



         % disp(' - THIRD model')

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         % disp(' -1-3')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
```

```
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         % disp(' -2-3 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
         end

        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
%          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          % disp('-3-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
           % disp('-4-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
          % disp(' -5-3')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
   Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
 if Ty(i,j)<0

           Ty(i,j) =Ty(i,j)*(-1);
```

```
        end
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end

        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%        - second model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          % disp('-6-3')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
if Vyp(i,j)<0
                         % disp('-6-3-
1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                     else Vyp(i,j)>0
                         % disp('-6-3-
2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
        % disp(' -7-3')
      Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
       if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
           Tm(i,j)=min
(Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       else

        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       end

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         % disp('-8-3 ')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0

           Tx(i,j) =Tx(i,j)*(-1);
       end
        if Ty(i,j)<0

          Ty(i,j) =Ty(i,j)*(-1);
```

```
               Tm(i,j)=min
(Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         else

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%          Ye(i,j)=y(i);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         end

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         % disp(' -9-3')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          % disp('-10-3')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%          - third model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         % disp(' -11-3')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          % disp('-12-3')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
```

```
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          % disp('-13-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0

            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


        % disp(' - FORTH model')

%         - first model
elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
           % disp(' -1-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);

         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
     elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
          % disp('-2-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          % disp('-3-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
         % disp(' -4-4 ')
          % disp('I dont know')
          444
           rr==1
           555
           return
           666


           elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          % disp('-5-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);




%        - second model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
         % disp('-6-4')

         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Xe(i,j)=x(j);
if Vyp(i,j)<0
                          % disp('-6-4-
1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                    else Vyp(i,j)>0
                         % disp(' -6-4-
2')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
         end
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
        % disp(' -7-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j)
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
           % disp('-8-4')

                     %
Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);

        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
           % disp('-9-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

            elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
           % disp('-10-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - third model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
        % disp('-11-4')
        3333

         Xe(i,j)=x(j+1);
         Ye(i,j)=y(i);
222
 rr==1
```

```
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
         % disp(' -12-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         % disp(' -13-4')
        % disp ('I dont know')
  rr==1


elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         % disp(' -1-5')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
%         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         % disp(' -2-5')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)>0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
         % disp(' -3-5')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
```

```
         % disp('-4-5')
         % disp('I dont know')
        rr==1

             elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
    rr==1

%      - second model
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
           % disp('-6-5')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%              Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
           Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%              Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)
        %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%          Xe(i,j)= x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
if Vyp(i,j)<0
                            % disp(' -6-5-
1')
          Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                    else Vyp(i,j)>0
                       % disp(' -6-5-
2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%              Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)
          %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
             elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
          % disp(' -7-5')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%              Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
          % disp(' -8-5 ')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%              Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
         % disp(' -9-5')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          % disp('-10-5')
                      Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j))
        %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          % disp('-11-5')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
%         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         % disp(' -12-5')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
    if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
%           Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
            % disp('-13-5 ')
            % disp('I dont know')

rr==1
%

elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
            % disp('-1-6')
            if ( Vxp(i,j)<0)
                % disp(' -1-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
            if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
            else ( Vxp(i,j)> 0)
                % disp('-1-6-2')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
            if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
            end
    elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
            % disp('-2-6')
            if ( Vxp(i,j)<0)
                % disp('-2-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
            else ( Vxp(i,j)> 0)
                % disp('-2-6-2')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
            Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
            Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
            end
    elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
            % disp('-3-6')
            if ( Vxp(i,j)<0)
                % disp('-3-6-1')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        if ( Vxp(i,j)<0)
```

```
            % disp(' -3-6-1')
      Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);
         else ( Vxp(i,j)> 0)
             % disp('-3-6-2')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
      Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
      Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end

       Tm(i,j)=Ty(i,j);
      Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);
         end
      elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
        % disp('-4-6')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        if ( Vxp(i,j)<0)
        % disp(' -4-6-1')
       Xe(i,j)=x(j);
      Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);
     else( Vxp(i,j)>0)
          % disp('-4-6-2')
          Xe(i,j)=x(j+1);
      Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);
     end


         elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
        % disp('-5-6')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
            if ( Vxp(i,j)<0)
         % disp('-5-6-1')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
      Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
       if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end

      Tm(i,j)=Ty(i,j);
```

```
         Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      else( Vxp(i,j)>0)
          % disp('-5-6-2')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

      Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      end



       elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
         % disp('-6-6')
         if Vyp(i,j)<0
             % disp(' -6-6-1')
              Ye(i,j)=y(i);
         else  Vyp(i,j)>0
              % disp('-6-6-2')
             Ye(i,j)=y(i+1);
         end
         if Vxp(i,j)<0
             % disp(' -6-6-3')
              Xe(i,j)=x(j);
         else  Vxp(i,j)>0
             % disp(' -6-6-4')
             Xe(i,j)=x(j+1);
         end

          elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
        % disp(' -7-6 ')
         if  Vxp(i,j)<0
             % disp(' -7-6-1')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end

       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
         else Vxp(i,j)>0
              % disp('-7-6-2')
```

```
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
          end

        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
         % disp(' -8-6')

          if  Vxp(i,j)<0
               % disp('-8-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          else Vxp(i,j)>0
               % disp('-8-6-2')
             Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
          end


        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
         % disp(' -9-6 ')
          if   Vxp(i,j)<0
              % disp('-9-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
          else Vxp(i,j)>0
             % disp(' -9-6-2')
            Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j)  ;
          end


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          % disp('-10-6')


Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          if   Vxp(i,j)<0
              % disp('-10-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
```

```
        Ye(i,j);
          else Vxp(i,j)>0
               % disp('-10-6-2')

        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
           end



%          - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
           % disp(' -11-6')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            if  Vxp(i,j)<0
             % disp(' -11-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          else  Vxp(i,j)>0
           % disp(' -11-6-2')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j) ;
          end

        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
           % disp(' -12-6')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            if  Vxp(i,j)<0
              % disp('-12-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          else  Vxp(i,j)>0
             % disp('-12-6-2')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
           end


        elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
           % disp('-13-6')


                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
```

```
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
           if  Vxp(i,j)<0
          % disp('-13-6-1 ')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);
      else  Vxp(i,j)>0
          % disp('-13-6-2 ')

        Xe(i,j)=x(j+1);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);
      end



      % disp(' -seventh model')

% - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
          % disp('-1-7')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
          % disp('-2-7 ')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          % disp('-3-7')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          % disp('-4-7 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j)
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
```

```
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          % disp('-5-7')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          % disp('-6-7')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         % disp(' -6-7-
1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
```

```
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                    else Vyp(i,j)>0
                        % disp('-6-7-
2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
         % disp(' -7-7 ')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         % disp(' -8-7 ')

               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          % disp('-9-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j)
        Ye(i,j)

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
         % disp(' -10-7 ')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
```

```
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

%          - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
          % disp(' -11-7')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%     Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          % disp(' -12-7')
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
```

```
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
        % disp(' -13-7')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j);
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


     % disp(' - eighth model')

%        - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         % disp('-1-8')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);



    elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         % disp('-2-8')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
         % disp('-3-8')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
         % disp(' -4-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          % disp('-5-8 ')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          % disp('-6-8')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
          if Vyp(i,j)<0
                         % disp('-6-8-
1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                      else Vyp(i,j)>0
                         % disp(' -6-8-
2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
         % disp(' -7-8 ')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1)
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         % disp(' -8-8 ')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          % disp('-9-8')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          % disp('-10-8 ')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%          - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         % disp(' -11-8')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          % disp(' -12-8')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
          elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          % disp('-13-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));


          if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=y(i);
          Xe(i,j);
          Ye(i,j);

          % disp(' -ninth model')
%           - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
          % disp(' -1-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
           Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
          Xe(i,j);
          Ye(i,j);


      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
          % disp(' -2-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
          end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j)
          Ye(i,j)
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          % disp('-3-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
            if Ty(i,j)<0
              Ty(i,j) =Ty(i,j)*(-1);
          end
          if Tx(i,j)<0
              Tx(i,j) =Tx(i,j)*(-1);
          end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          % disp('-4-9 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          % disp('-5-9')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
```

```
          % disp('-6-9')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         % disp('-6-9-
1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                     else Vyp(i,j)>0
                         % disp('-6-9-
2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
          % disp('-7-9')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
```

```
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         % disp(' -8-9 ')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         % disp(' -9-9')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          % disp('-10-9')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

%        - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         % disp(' -11-9')

                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
```

```
Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
        % disp(' -12-9')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
        % disp(' -13-9 ')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


       % disp(' -tenth  model')
%        - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         % disp('-1-10')
```

```
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         % disp(' -2-10')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)<0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          % disp('-3-10 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
```

```
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
          % disp('-4-10')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
%          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         % disp(' -5-10')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
```

```
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%         - second model
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
         % disp(' -6-10')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                      % disp(' -6-10-
1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                   else Vyp(i,j)>0
                      % disp(' -6-10-
2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
```

```
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
          % disp('-7-10 ')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
         % disp(' -8-10')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
          % disp('-9-10')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          % disp('-10-10')
                        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
```

```
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          % disp('-11-10')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
          % disp('-12-10')
               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
         % disp(' -13-10 ')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        % disp(' -eleventh model')

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         % disp(' -1-11')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)==0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         % disp(' -2-11')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
     elseif  (Vx(i,j)==0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          % disp('-3-11')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
          % disp('-4-11')
          % disp('I dont know')
 Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%               Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         % disp(' -5-11')
           % disp('I dont know')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
             Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%        - second model
```

```
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
         % disp(' -6-11')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                          % disp(' -6-11-
1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Xe(i,j);
        Ye(i,j);

                       else Vyp(i,j)>0
                          % disp(' -6-11-
2')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%           Xe(i,j)= x(j+1)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
          % disp('-7-11')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
          % disp('-8-11')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
          % disp('-9-11')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
          Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
           if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
         Xe(i,j);
         Ye(i,j);

            elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
           % disp('-10-11')
                       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
          Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);



%          - third model
         elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          % disp(' -11-11 ')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j)
```

```
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
        % disp(' -12-11')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         % disp('-13-11')
        % disp(' I dont know')

rr==1



 % disp('-TWELVE MODEL')
```

```
elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
        % disp(' -1-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        % disp(' -2-12')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

      elseif  (Vx(i,j)==0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
        % disp(' -3-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
         % disp('-4-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1))
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


          elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
        % disp(' -5-12')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i))+Vy(i,j);
%           Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

%       - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
          % disp('-6-12')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i))+Vy(i,j);
        if Vyp(i,j)<0
                        % disp('-6-12-
1')
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                    else Vyp(i,j)>0
                        % disp(' -6-12-
2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
          Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
        end
          elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
          % disp(' -7-12')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
          % disp(' -8-12')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
          % disp('-9-12')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          % disp('-10-12')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
%          - third model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          % disp('-11-12 ')

                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         % disp(' -12-12')
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(+1j)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         % disp(' -13-12')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        % disp(' - thirtheenth  model')



          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)>0
& Vy(i,j)==Vy(i+1,j))
         % disp(' -1-13')
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)==0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         % disp(' -2-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)==0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          % disp('-3-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
          % disp('-4-13')
         % disp('Idont know')
   rr==1

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          % disp('-5-13')
         % disp(' I dont know')

      rr==1


%        - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
          % disp('-6-13 ')

         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                             % disp('-6-13-
1')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                         else Vyp(i,j)>0
                             % disp('-6-13-
2')
         Xe(i,j)=x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
```

```
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
         % disp('-7-13')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
        % disp(' -8-13')

       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
         % disp('-9-13')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);
       Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);

          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
         % disp('-10-13')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);



%        - third model
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
        % disp(' -11-13 ')

       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
        % disp(' -12-13')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
          elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
         % disp('-13-13 ')
         % disp('I dont know')
```

```
rr==1
%
            else
                % disp('new model')

                Vx(i,j);
                 Vx(i,j+1);
                  Vy(i,j);
                   Vy(i+1,j);
            end

end

%    Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM

%    Developed MATLAB Program for
Streamline Simulation
%    This Code developed originally by
MARJAN HASHEM

%    Second Case Study in Cartesian
Coordinate
%    Subroutine for Finding the Pressure
Distribution all Across the Grid Blocks

function pressure
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B
P=zeros(Y,X);

    maxr=1;
errormatrix=zeros(X,Y);
itteration=0;
while maxr>maxerr
    for i=1:Y
        for j=2:X-1
            tempval=P(i,j);
            x=1;
            y=1;


        if i==1
            1;
          P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i+1,j))/4);

        elseif i==Y
            2;
          P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i-1,j))/4);


         elseif  (i==fix(Y/3)&
j<=fix((2*X)/3) & j>=fix((X)/3))
            3;
                                    4;

            P(i,j)=(1/4)*(P(i,j+1)+
(((2*K1)/(K1+K2))*P(i+1,j))+P(i,j-1)+
(((2*K2)/(K1+K2))*P(i-1,j)));
            %               end

                    elseif
i==(fix((2*Y)/3)& j<=fix((2*X)/3) &
j>=fix((X)/3))
%              elseif  i==(fix((2*Y)/3)&
j<fix((2*X)/3) & j>fix((X)/3))
                5;
%                 if i==fix(Y/3)
     %               if
(j<fix((2*X)/3) & j>fix((X)/3))
                6;

            P(i,j)=(1/4)*(P(i,j+1)+
(((2*K1)/(K1+K2))*P(i-1,j))+P(i,j-1)+
(((2*K2)/(K1+K2))*P(i+1,j)));
            %               end

            elseif  (j== fix(X/3)&
i>fix(Y/3) & i<fix((2*Y)/3) )
%            elseif  (j== fix(X/3)&
i>=fix(Y/3) & i<=fix((2*Y)/3) )

      %              if i>=fix(Y/3) &
i<=fix((2*Y)/3)

            P(i,j)=(1/4)*(P(i-1,j)+
(((2*K1)/(K1+K2))*P(i,j+1))+P(i+1,j)+
(((2*K2)/(K1+K2))*P(i,j-1)));
            %               end

            elseif  (j==
fix((2*X)/3)& i>fix(Y/3) &
i<fix((2*Y)/3))
%                elseif  (j==
fix((2*X)/3)& i>=fix(Y/3) &
i<=fix((2*Y)/3))

      %              if i>=fix(Y/3) &
i<=fix((2*Y)/3)

            P(i,j)=(1/4)*(P(i-1,j)+
(((2*K1)/(K1+K2))*P(i,j-1))+P(i+1,j)+
(((2*K2)/(K1+K2))*P(i,j+1)));
            %               end
            else

P(i,j)=((P(i,j+1)+P(i,j-
1)+P(i+1,j)+P(i-1,j))/4);
            end


errormatrix(i,j)=abs(P(i,j)-tempval);
         end
    end
    maxr=max(max(errormatrix));
    itteration=itteration+1;
end
```

```
for i=1:Y
        for j=1:X
            P(i,j);
        end
    end

        contourf (P,50, 'DisplayName',
'PRESSURE');colormap autumn;
        figure
itteration;
 P;

%   Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   Subroutine for drawing the
Streamline

global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
temp = find(Xp);
xx = Xp(temp)';
yy = Yp(temp)';
  xx = [Xp(1,1); xx];
  yy = [Yp(1,1); yy];
plot(xx,yy,'*-')
grid on
axis equal
axis square
%   Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   Subroutine for Drawing the
Streamline in "x" Direction
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
temp = find(Xp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'*-')
grid on
axis equal
axis square
%   Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Cartesian
Coordinate
%   Subroutine for Drawing the
Streamline in "y" Direction
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B
temp = find(Yp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'*-')
grid on
axis equal
axis square
```

Third Case Study in Cartesian Coordinate

```
%   Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM
```

```
%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
%   Main route
clc;
clear all;
close all;
format long e

Y = input('Number of Nodes in Y ');
X = input('Number of Nodes in X ');

% Boundary values can be set along the
corner or in the middle
reply = input('Corner or Middle
boundary conditions C/M [C]: ', 's');

%Maximum Error abs(x2-x1)
maxerr= input('Desired Maximum Error
[.00001]: ');
if isempty(reply)
    reply = 'C';
end
if isempty(maxerr)
    maxerr = .00001;
end

global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B

permeabilities


reply=lower(reply);
if reply=='c'
pressure
else
    for i=floor(X/3):floor(2*X/3)
    P(i,1)=(2*i);
end
end


M=(0.5*10^-3);
velocities

for b=1:X+1
    x(b)=(b-1);
end
for t=1:Y+1
    y(t)=(t-1);
end
    Xp=zeros(X,Y);
```

```
    Yp=zeros(X,Y);


     Vxp=zeros(X,Y);
    Vyp=zeros(X,Y);



123456789
                    for m=2:Y
Xp(m,1)=0;
 Yp(m,1)=(m-1);
  Xpp=Xp(m,1);
  Ypp=Yp(m,1) ;
 Vxp(m,1)=Vx(m,1);
 Vyp(m,1)=Vy(m,1) ;
  i=m;
j=1;

            while ((Xpp<(X-1) ) & (Ypp<Y)
& (i<(Y)) & (j<(X-1)))
%               while ((Xpp<(X-1) ) &
(Ypp<Y) )

 location ;
888
if rr==1;
    999;
    break
end
1010
% input('2')
if (Ye(i,j)>=0 & Xe(i,j)>=0)
    1111
    Xpp
    Ypp
    Xe(i,j)
    Ye(i,j)
    x(j)
    y(i)
    i
    j
            if (Xe(i,j)-(fix(Xe(i,j))))==0
        1231
        B=Xe(i,j)+1
        elseif ((Xe(i,j)-
(fix(Xe(i,j))))>=0.9 & (Xe(i,j)-
(fix(Xe(i,j))))<1)
        4561
                B=(round(Xe(i,j)))+1
        elseif ((Xe(i,j)-
(fix(Xe(i,j))))>0 & (Xe(i,j)-
(fix(Xe(i,j))))<=0.01)
                B=(round(Xe(i,j)))+1
        else
        B=(fix(Xe(i,j)))+1
            end
            B

            if (Ye(i,j)-(fix(Ye(i,j))))==0
            1232
                A=Ye(i,j)+1
            elseif ((Ye(i,j)-
(fix(Ye(i,j))))>0.9 & (Ye(i,j)-
(fix(Ye(i,j))))<1)
                4562
                A=(round(Ye(i,j)))+1
                elseif ((Ye(i,j)-
(fix(Ye(i,j))))>0 & (Ye(i,j)-
(fix(Ye(i,j))))<=0.01)
```

```
            A=(round(Ye(i,j)))+1
        else
          A=(fix(Ye(i,j)))+1
        end
          A

            Xp(A,B)=Xe(i,j);
            Yp(A,B)=Ye(i,j);
            Xpp=Xp(A,B);
            Ypp=Yp(A,B);
              i=A;
            j=B;

                rr==0;

else

    rr==1;
     break

end



          x(j);



    end
        plotFinalforYdirection
        hold on
        Xp=zeros(X,Y);
        Yp=zeros(X,Y);
       1731317
     end

%   Streamline Simulation Near Well
Bore
%        By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
%   Subroutine for finding Permeability

function permeabilities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp

    for i=1:Y
    for j=1:X
        Kx(i,j)=(0.2*(10^-2));
         Ky(i,j)=(0.2*(10^-2));
    end
```

```
      end
      Kx
      Ky

%   Streamline Simulation Near Well
Bore
%        By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
%   Subroutine for finding the
velocity
function velocities
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp

M=(0.5*10^-3);
  Vx=zeros(Y+1,X+1);
  Vy=zeros(Y+2,X);
    for i=1:(Y+1)

      Vx(i,1)=(0.005) ;




end
for i=2:Y+1
      for j=2:X
   Vx(i-1,j)=(-1)*((Kx(i-1,j-1)*(P(i-
1,j)-P(i-1,j-1)))/(M*x));

       end
end
for i=Y+1
      for j=1:(X+1)
         Vx(i,j)=Vx(i-1,j);
      end
end





 for i=2:Y
            for j=2:X+1
Vy(i,j-1)=(-1)*((Ky(i-1,j-1)*(P(i,j-1)-
P(i-1,j-1)))/(M*y));
```

```
    end
  end

Vx;
 Vy;


% Streamline Simulation Near Well
Bore
%           By MARJAN HASHEM

% Developed MATLAB Program for
Streamline Simulation
% This Code developed originally by
MARJAN HASHEM

% Third Case Study in Cartesian
Coordinate
% Subroutine for finding Pressure
distribution all across the grid blocks
function pressure
global X Y
global Kx Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B

P=zeros(Y,X);

 for i=1:Y

        P(i,1)=100000;

              for
i=(fix((7*Y)/10)):(fix((9*Y)/10))

                 P(i,X)=100;


              end

 end



      maxr=1;
errormatrix=zeros(X,Y);
itteration=0;
while maxr>maxerr
      for i=1:Y
          for j=2:X
              tempval=P(i,j);
              x=1;
              y=1;


          if (i==1 & j<=(X-1))
                    1;
                  P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i+1,j))/4);

               elseif (i==Y & j<=(X-1))
                    2;

                  P(i,j)=((P(i,j-
1)+P(i,j+1)+2*P(i-1,j))/4);

               elseif (j==X & i==1)

                   P(i,j)=((P(i+1,j)+P(i,j-
1))/2);

               elseif(j==X & i==Y)

                   P(i,j)=((P(i-
1,j)+P(i,j-1))/2);

               elseif (j==X & i>1 &
i<fix((7*Y)/10))
                    3

                    P(i,j)=((P(i-
1,j)+P(i+1,j)+(2*P(i,j-1)))/4);

                 elseif (j==X &
i>=(fix((9*Y)/10)+1) & i<Y)
                    4

                    P(i,j)=((P(i-
1,j)+P(i+1,j)+(2*P(i,j-1)))/4);
                 elseif (j==X &
i>=(fix((7*Y)/10)) &
i<=(fix((9*Y)/10)))
                    5
                    P(i,j)=100;


                   else
                    11;

P(i,j)=((P(i,j+1)+P(i,j-
1)+P(i+1,j)+P(i-1,j))/4);
              end


errormatrix(i,j)=abs(P(i,j)-tempval);
          end
      end
      maxr=max(max(errormatrix));
      itteration=itteration+1;
end

for i=1:Y
        for j=1:X
            P(i,j);
        end
    end


        contourf (P,50, 'DisplayName',
'PRESSURE');colormap autumn;
        figure
itteration
P
```

```
%   Streamline Simulation Near Well
Bore
%         By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
% Subroutine for finding the Exit
location for streamline's particle

function location
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg ff
global i j
global rr
global Ax Ay Vxp Vyp
global K1 K2
global A B
% fid2 = fopen('any name.dat','w');


disp ('first model')
 9876
if  (Vx(i,j)> 0 & Vx(i,j+1)> 0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)> 0 &
Vy(i+1,j)> 0 & Vy(i,j)==Vy(i+1,j))
      disp(' 1-1 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
       if Ty(i,j)<0
           1317
         Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           1318
           Tx(i,j) =Tx(i,j)*(-1);
       end

       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);
 elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
       disp(' 2-1 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            1317
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           1318
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
        disp(' 3-1 ')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
            1317
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           1318
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 );
         disp('-4-1')
```

```
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
           %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
if Ty(i,j)<0
    1317
        Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


    elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
        disp('-5-1')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
             1317
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-1')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
%         Xe(i,j)= x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
  if Vyp(i,j)<0
                               disp('-6-1-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                           else Vyp(i,j)>0
                               disp('-6-1-2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
          disp('-7-1')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1317
            Tx(i,j) =Tx(i,j)*(-1);
```

```
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
          disp('-8-1')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          disp('-9-1')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
              1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
          disp('-10-1')
                      Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%         - third model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-1')


                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
```

```
            1317
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
           disp('-12-1')
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
          if Ty(i,j)<0
              1317
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
           disp('-13-1')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
```

```
        end
        Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



         disp(' - second model')

%          - first model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
          disp('-1-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
              1317
            Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);



elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp(' -2-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0
            1317
            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0
             1317
            Ty(i,j) =Ty(i,j)*(-1);
         end

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
          disp('-5-2')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
             1317
          Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
         disp(' -6-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
if Vyp(i,j)<0
                            disp('-6-2-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                        else Vyp(i,j)>0
                            disp('-6-2-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
         disp('-7-2')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        else

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp('-8-2')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i,j));
 if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0
             1317
               Ty(i,j) =Ty(i,j)*(-1);
         end


         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         disp('-9-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
              1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)==0  )
         disp(' -10-2')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0
             1317
            Ty(i,j) =Ty(i,j)*(-1);
         end
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));
           %         Tm(i,j)=min
(Tx(i,j),Ty(i,j))
%          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%          Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



%          - third model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp('-11-2')

               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
          if Tx(i,j)<0
              1318
             Tx(i,j) =Tx(i,j)*(-1);
        end
 if Ty(i,j)<0
     1317
            Ty(i,j) =Ty(i,j)*(-1);
 end

         Tm(i,j)=min (Tx(i,j),Ty(i,j));
%          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
%  end
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
         disp(' -12-2')
                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
     1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)==0  )
         disp(' -13-2')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


         disp(' - THIRD model')

%          - first model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp(' -1-3')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
              1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp(' -2-3 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));
%               Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         else

        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)==0 )
         disp(' -5-3')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
    Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
 if Ty(i,j)<0
      1317
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end

        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%         - second model
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-3')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
if Vyp(i,j)<0
                        disp('-6-3-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                    else Vyp(i,j)>0
                        disp('-6-3-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j+1);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)> Vy(i+1,j) )
         disp(' -7-3')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
            Tm(i,j)=min
(Tx(i,j),Ty(i,j));
%              Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* 0))-Vy(i,j)))+y(i);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%           Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
        else

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%          Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
        end

elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp('-8-3 ')

                        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
```

```
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
         if Ty(i,j)<0
             1317
           Ty(i,j) =Ty(i,j)*(-1);
             Tm(i,j)=min
(Tx(i,j),Ty(i,j));
%               Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
%           Ye(i,j)=y(i);
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* 0)))-Vy(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
         else

         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%          Ye(i,j)=y(i);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         end


elseif  (Vx(i,j)>0 & Vx(i,j+1)>0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         disp(' -9-3')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          disp('-10-3')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);



%          - third model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-3')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 & Vx(i,j+1)>0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
          disp('-12-3')
                      Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)>0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          disp('-13-3')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            1318
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);



        disp(' - FORTH model')

%          - first model
```

```
elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
            disp(' -1-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
          disp('-2-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)>0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          disp('-3-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
         disp(' -4-4 ')
          disp('I dont know')
          444
```

```
            rr==1
            555
            return
            666

% return


            elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          disp('-5-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%        - second model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
         disp('-6-4')

         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Xe(i,j)=x(j);
if Vyp(i,j)<0
                         disp('-6-4-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                     else Vyp(i,j)>0
                        disp(' -6-4-2')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
         disp(' -7-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j)
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
          disp('-8-4')

                    %
Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
          disp('-9-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
```

```
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%          - third model
        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
          disp('-11-4')
          3333

         Xe(i,j)=x(j+1);
         Ye(i,j)=y(i);
%         Xe(i,j);
%         Ye(i,j);
222
 rr==1


        elseif  (Vx(i,j)>0 & Vx(i,j+1)<0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
         disp(' -12-4')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         disp(' -13-4')
        disp ('I dont know')
  rr==1



 disp('fifth model')

elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         disp(' -1-5')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
%          Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)>0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         disp(' -2-5')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)>0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
         disp(' -3-5')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
          disp('-4-5')
          disp('I dont know')
        rr==1

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          disp('-5-5')
           disp('I dont know')
    rr==1


        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
          disp('-6-5')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

if Vyp(i,j)<0
                            disp(' -6-5-1')
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                         else Vyp(i,j)>0
                            disp(' -6-5-2')
%          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)
        %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
         disp(' -7-5')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-5 ')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
         disp(' -9-5')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-5')
                        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j))
        %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%         - third model
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-5')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
%        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
```

```
        elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         disp(' -12-5')
               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       %         Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)>0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
          disp('-13-5 ')
          disp('I dont know')

rr==1



 disp('-sixth  model')


elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
          disp('-1-6')
           if ( Vxp(i,j)<0)
              disp(' -1-6-1')
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
           else ( Vxp(i,j)> 0)
                disp('-1-6-2')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
           end
   elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
          disp('-2-6')
           if ( Vxp(i,j)<0)
                disp('-2-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
            else ( Vxp(i,j)> 0)
             disp('-2-6-2')
              Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
            end
      elseif  (Vx(i,j)<0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
```

```
            disp('-3-6')
             if ( Vxp(i,j)<0)
                  disp('-3-6-1')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
            if ( Vxp(i,j)<0)
                 disp(' -3-6-1')
          Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             else ( Vxp(i,j)> 0)
                 disp('-3-6-2')
              Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

           Tm(i,j)=Ty(i,j);
          Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
          Xe(i,j);
          Ye(i,j);
             end
          elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)<0 )
            disp('-4-6')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
            if ( Vxp(i,j)<0)
            disp(' -4-6-1')
           Xe(i,j)=x(j);
          Ye(i,j)=y(i);
           Xe(i,j);
           Ye(i,j);
         else( Vxp(i,j)>0)
              disp('-4-6-2')
              Xe(i,j)=x(j+1);
          Ye(i,j)=y(i);
           Xe(i,j);
           Ye(i,j);
         end
```

```
               elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
             disp('-5-6')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
                Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
                 if ( Vxp(i,j)<0)
              disp('-5-6-1')
           Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
           Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

          Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
           Xe(i,j);
           Ye(i,j);
         else( Vxp(i,j)>0)
             disp('-5-6-2')
              Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
           Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
           if Ty(i,j)<0
             Ty(i,j) =Ty(i,j)*(-1);
          end

         Tm(i,j)=Ty(i,j);
            Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
           Xe(i,j);
           Ye(i,j);
         end



          elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)>0 )
            disp('-6-6')
            if Vyp(i,j)<0
                disp(' -6-6-1')
                 Ye(i,j)=y(i);
            else  Vyp(i,j)>0
                 disp('-6-6-2')
                Ye(i,j)=y(i+1);
            end
            if Vxp(i,j)<0
                disp(' -6-6-3')
                 Xe(i,j)=x(j);
            else  Vxp(i,j)>0
                disp(' -6-6-4')
                Xe(i,j)=x(j+1);
            end
             elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
           disp(' -7-6 ')
            if  Vxp(i,j)<0
```

```
            disp(' -7-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        else Vxp(i,j)>0
          disp('-7-6-2')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
        end

      elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)<
Vy(i+1,j) )
        disp(' -8-6')

        if  Vxp(i,j)<0
          disp('-8-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        else Vxp(i,j)>0
          disp('-8-6-2')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
        end

      elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)==
Vy(i+1,j) )
        disp(' -9-6 ')
        if  Vxp(i,j)<0
          disp('-9-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);
        else Vxp(i,j)>0
          disp(' -9-6-2')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j)  ;
        end

      elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
        disp('-10-6')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
        if  Vxp(i,j)<0
            disp('-10-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          else Vxp(i,j)>0
              disp('-10-6-2')

        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
          end


%         - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)>0  )
         disp(' -11-6')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
           if  Vxp(i,j)<0
            disp(' -11-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         else  Vxp(i,j)>0
            disp(' -11-6-2')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j) ;
          end

        elseif  (Vx(i,j)<0 & Vx(i,j+1)>0
& Vy(i,j)==0 & Vy(i+1,j)<0  )
         disp(' -12-6')

                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          if  Vxp(i,j)<0
               disp('-12-6-1')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
           else  Vxp(i,j)>0
               disp('-12-6-2')
                Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end

        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j+1);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
          end


        elseif  (Vx(i,j)<0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
         disp('-13-6')


             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
          if  Vxp(i,j)<0
         disp('-13-6-1 ')
% %         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Xe(i,j)=x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
     else  Vxp(i,j)>0
         disp('-13-6-2 ')

       Xe(i,j)=x(j+1);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);
     end



     disp(' -seventh model')

% - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp('-1-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
```

```
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp('-2-7 ')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
         disp('-3-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
          disp('-4-7 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j)
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          disp('-5-7')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
           Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);




%       - second model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         disp(' -6-7-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                      else Vyp(i,j)>0
                         disp('-6-7-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
         disp(' -7-7 ')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-7 ')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
          disp('-9-7')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j)
        Ye(i,j)

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
         disp(' -10-7 ')
                       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));
        Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

%         - third model
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-7')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
%    Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)==Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
        disp(' -12-7')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)==Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
        disp(' -13-7')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j);
       Xe(i,j)=(Tm(i,j)*
Vxp(i,j))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);


     disp(' - eighth model')

%        - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp('-1-8')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp('-2-8')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
          disp('-3-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
         disp(' -4-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


          elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
         disp('-5-8 ')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
          disp('-6-8')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          if Vyp(i,j)<0
                         disp('-6-8-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                        else Vyp(i,j)>0
                           disp(' -6-8-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
```

```
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
       disp(' -7-8 ')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
      Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
      Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
      Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
      Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
      Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end
      if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1)
      end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);
      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
       disp(' -8-8 ')

                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
      Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
      Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
      Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
      Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
      Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
       if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end
      if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
      end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);

      elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
        disp('-9-8')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
      Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
      Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
      Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
      Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end
      if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
      end
      Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
      Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
      Xe(i,j);
      Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
       disp('-10-8 ')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
      Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
      Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
      Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
      Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
      Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
       if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end
      if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
      end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);



%       - third model
```

```
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
         disp(' -11-8')

                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)>Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
         disp(' -12-8')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)>Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
          disp('-13-8')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

        disp(' -ninth model')
%          - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)==Vy(i+1,j))
         disp(' -1-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)>Vy(i+1,j))
         disp(' -2-9')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
```

```
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
     elseif  (Vx(i,j)<0 & Vx(i,j+1)<0 &
Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)>0 & Vy(i,j)<Vy(i+1,j))
         disp('-3-9')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)>0 &
Vy(i+1,j)<0 )
         disp('-4-9 ')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);
```

```
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)>0 & Vy(i+1,j)==0 )
          disp('-5-9')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);



%        - second model
       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)>0 )
         disp('-6-9')

       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        if Vyp(i,j)<0
                          disp('-6-9-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
         Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);
```

```
                    else Vyp(i,j)>0
                        disp('-6-9-2')
          Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)<0 &  Vy(i,j)>
Vy(i+1,j) )
          disp('-7-9')
                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-9 ')

                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)<0 &
Vy(i+1,j)<0 &  Vy(i,j)== Vy(i+1,j) )
         disp(' -9-9')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)<0 & Vy(i+1,j)==0  )
          disp('-10-9')

Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
```

```
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);




%        - third model
       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)>0  )
        disp(' -11-9')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
          Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)<0 & Vx(i,j+1)<0
& Vx(i,j)<Vx(i,j+1)& Vy(i,j)==0 &
Vy(i+1,j)<0  )
        disp(' -12-9')
                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j)
       Ye(i,j)
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)<0 & Vx(i,j)<Vx(i,j+1)&
Vy(i,j)==0 & Vy(i+1,j)==0  )
        disp(' -13-9 ')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
       Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);




       disp(' -tenth  model')
%        - first model
elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         disp('-1-10')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));
```

```
Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)<0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         disp(' -2-10')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
     elseif  (Vx(i,j)<0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          disp('-3-10 ')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
          disp('-4-10')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         %
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


            elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         disp(' -5-10')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        %          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);




%        - second model
```

```
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
         disp(' -6-10')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                            disp(' -6-10-1')
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


                      else Vyp(i,j)>0
                            disp(' -6-10-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
 %          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j))*(-1);
%          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i))
%          Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j)
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
          disp('-7-10 ')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-10')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
          disp('-9-10')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
```

```
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-10')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);




%         - third model
        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-10')
                    Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
          disp('-12-10')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
 if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)<0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
         disp(' -13-10 ')
```

```
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j)-
Xp(i,j))/Vx(i,j));

       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=y(i);
       Xe(i,j);
       Ye(i,j);




       disp(' -eleventh model')

       elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
        disp(' -1-11')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);


   elseif  (Vx(i,j)==0 & Vx(i,j+1)>0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
        disp(' -2-11')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
     elseif  (Vx(i,j)==0 & Vx(i,j+1)>0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
         disp('-3-11')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
         disp('-4-11')
         disp('I dont know')
 Ax(i,j)=(Vx(i,j+1)-Vx(i,j))/(x(j+1)-
x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%               Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
```

```
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
          Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         disp(' -5-11')
           disp('I dont know')
              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
             Vxp(i,j)=
(Ax(i,j)*(Xp(i,j)-x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);






%        - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
         disp(' -6-11')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
         if Vyp(i,j)<0
                             disp(' -6-11-1')
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
%         Xe(i,j)= x(j)
%
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i)
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                         else Vyp(i,j)>0
                             disp(' -6-11-2')
            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
          disp('-7-11')
                  Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
```

```
        Xe(i,j)
        Ye(i,j)
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
          disp('-8-11')

             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
          disp('-9-11')
        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-11')
                     Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

%         - third model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
         disp(' -11-11 ')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
%         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j)
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));
```

```
Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
         disp(' -12-11')
               Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=(x(j+1)-
Xp(i,j))/Vx(i,j+1);
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
         elseif  (Vx(i,j)==0 &
Vx(i,j+1)>0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
          disp('-13-11')
         disp(' I dont know')

rr==1
%


 disp('-TWELVE MODEL')
elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)==Vy(i+1,j))
         disp(' -1-12')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);


   elseif  (Vx(i,j)==0 & Vx(i,j+1)<0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
         disp(' -2-12')
           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
         Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)==0 & Vx(i,j+1)<0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
         disp(' -3-12')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
```

```
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)<0 )
          disp('-4-12')
         Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1))

        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
         Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
        Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);


           elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
         disp(' -5-12')
             Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
            Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%
Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i)-Yp(i,j))/Vy(i,j);
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
      Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);


%        - second model
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)>0 )
          disp('-6-12')

        Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
         Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                         disp('-6-12-1')
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                     else Vyp(i,j)>0
                        disp(' -6-12-2')
%           Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
        Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j));
        if Tx(i,j)<0
            Tx(i,j) =Tx(i,j)*(-1);
        end
       Tm(i,j)=Tx(i,j)

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i+1);
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)> Vy(i+1,j) )
         disp(' -7-12')
                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
```

```
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);
       elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)< Vy(i+1,j) )
        disp(' -8-12')

            Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
       Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
       Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

       elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)<0 &
Vy(i,j)== Vy(i+1,j) )
         disp('-9-12')
       Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
        Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
```

```
%          Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
       Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
       Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
       Xe(i,j);
       Ye(i,j);

         elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
        disp('-10-12')
                   Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
       Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%         Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
        Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
        if Ty(i,j)<0
          Ty(i,j) =Ty(i,j)*(-1);
       end
       if Tx(i,j)<0
           Tx(i,j) =Tx(i,j)*(-1);
       end
        Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);




%        - third model
```

```
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
          disp('-11-12 ')

                 Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%          Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j)
 if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);

         elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
          disp(' -12-12')
                Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(+1j)-
Xp(i,j))/Vx(i,j+1));
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
%          Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j))*(-1)
 if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=min (Tx(i,j),Ty(i,j));

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
         Xe(i,j);
         Ye(i,j);
          elseif  (Vx(i,j)==0 &
Vx(i,j+1)<0 & Vy(i,j)==0 & Vy(i+1,j)==0
)
          disp(' -13-12')

              Ax(i,j)=(Vx(i,j+1)-
Vx(i,j))/(x(j+1)-x(j));
          Vxp(i,j)= (Ax(i,j)*(Xp(i,j)-
x(j)))+Vx(i,j);
%           Tx(i,j)=(1/Ax(i,j))*log
(Vx(i,j+1)/Vxp(i,j))
          Tx(i,j)=((x(j+1)-
Xp(i,j))/Vx(i,j+1));
         if Tx(i,j)<0
             Tx(i,j) =Tx(i,j)*(-1);
         end
         Tm(i,j)=Tx(i,j);

Xe(i,j)=((1/Ax(i,j))*((Vxp(i,j)*exp(Ax(
i,j)* Tm(i,j)))-Vx(i,j)))+x(j);
         Ye(i,j)=y(i);
         Xe(i,j);
         Ye(i,j);



         disp(' - thirtheenth  model')



           elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)>0
& Vy(i,j)==Vy(i+1,j))
          disp(' -1-13')
          Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%          Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i,j);
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j);
         Ye(i,j)=(Tm(i,j)*
Vxp(i,j))+y(i);
         Xe(i,j);
         Ye(i,j);


    elseif  (Vx(i,j)==0 & Vx(i,j+1)==0 &
Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)>Vy(i+1,j))
          disp(' -2-13')
         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
          if Ty(i,j)<0
            Ty(i,j) =Ty(i,j)*(-1);
         end
         Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j);
```

```
Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
      elseif  (Vx(i,j)==0 & Vx(i,j+1)==0
& Vy(i,j)>0 & Vy(i+1,j)>0 &
Vy(i,j)<Vy(i+1,j))
          disp('-3-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
         Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)<0
)
          disp('-4-13')
         disp('Idont know')
   rr==1

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)>0 & Vy(i+1,j)==0
)
          disp('-5-13')
         disp(' I dont know')

      rr==1



        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)>0
)
          disp('-6-13 ')

         Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
         Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
         if Vyp(i,j)<0
                          disp('-6-13-1')
        Xe(i,j)=x(j);
         Ye(i,j)=y(i);
        Xe(i,j);
        Ye(i,j);

                      else Vyp(i,j)>0
                          disp('-6-13-2')
        Xe(i,j);
        Ye(i,j);
          end
           elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)> Vy(i+1,j) )
          disp('-7-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)< Vy(i+1,j) )
         disp(' -8-13')

        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j));
         if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
         Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
        Xe(i,j);
        Ye(i,j);

        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)<0
&  Vy(i,j)== Vy(i+1,j) )
          disp('-9-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
         Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
          if Ty(i,j)<0
           Ty(i,j) =Ty(i,j)*(-1);
        end
        Tm(i,j)=Ty(i,j);
        Xe(i,j)=x(j);
        Ye(i,j)=(Tm(i,j)*
Vyp(i,j))+y(i);
        Xe(i,j);
        Ye(i,j);

           elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)<0 & Vy(i+1,j)==0
)
          disp('-10-13')
        Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
        Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%         Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=((y(i)-
Yp(i,j))/Vy(i,j));
         if Ty(i,j)<0
```

```
        Ty(i,j) =Ty(i,j)*(-1);
      end
       Tm(i,j)=Ty(i,j);
       Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);





      elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)>0
)
       disp(' -11-13 ')

       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
        Ty(i,j)=(y(i+1)-
Yp(i,j))/Vy(i+1,j);
       if Ty(i,j)<0
        Ty(i,j) =Ty(i,j)*(-1);
      end
      Tm(i,j)=Ty(i,j);
      Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
       Xe(i,j);
       Ye(i,j);

      elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 & Vy(i+1,j)<0
)
       disp(' -12-13')
       Ay(i,j)=(Vy(i+1,j)-
Vy(i,j))/(y(i+1)-y(i));
       Vyp(i,j)=(Ay(i,j)*(Yp(i,j)-
y(i)))+Vy(i,j);
%        Ty(i,j)=(1/Ay(i,j))*log
(Vy(i+1,j)/Vyp(i,j))
       Ty(i,j)=((y(i+1)-
Yp(i,j))/Vy(i+1,j));
        if Ty(i,j)<0
         Ty(i,j) =Ty(i,j)*(-1);
      end
      Tm(i,j)=Ty(i,j);
      Xe(i,j)=x(j);

Ye(i,j)=((1/Ay(i,j))*((Vyp(i,j)*exp(Ay(
i,j)* Tm(i,j)))-Vy(i,j)))+y(i);
      Xe(i,j);
      Ye(i,j);
        elseif  (Vx(i,j)==0 &
Vx(i,j+1)==0 & Vy(i,j)==0 &
Vy(i+1,j)==0  )
       disp('-13-13 ')
       disp('I dont know')

rr==1

        else
```

```
            disp('new model')

            Vx(i,j)
             Vx(i,j+1)
              Vy(i,j)
               Vy(i+1,j)
       end
       5755823
% fprintf(fid2,' Xe     Ye      Ax
Vxp    Ay     Vyp /L\n\n');
%  fprintf(fid2,'%f  %f  %e   %e  %e
%e\n',Xe,Ye,Ax,Vxp,Ay,Vyp);
 end

%   Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
%   Subroutine for drawing streamline
in "x" direction
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
temp = find(Xp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'*-')
grid on
axis equal
axis square
%   Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Third Case Study in Cartesian
Coordinate
%   Subroutine for drawing streamline
in "y" direction

global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
```

```
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
temp = find(Yp~=0);
xx = Xp(temp)';
yy = Yp(temp)';
xx = [xx];
yy = [yy];
plot(xx,yy,'*-')
grid on
axis equal
axis square
%    Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%    Developed MATLAB Program for
Streamline Simulation
%    This Code developed originally by
MARJAN HASHEM

%    Third Case Study in Cartesian
Coordinate
%    Subroutine for drawing streamline
global X Y
global Kx  Ky
global P
global Vx Vy
global Xe Ye
global maxerr maxr errormatrix
global x y
global Xpp Ypp
global Xp Yp
global Vxp Vyp
global gg
global i j
global rr
global Ax Ay Vxp Vyp
% temp = find(Xp~=0);
temp = find(Xp);
xx = Xp(temp)';
yy = Yp(temp)';
  xx = [Xp(1,1); xx];
  yy = [Yp(1,1); yy];
%  xx = [0.5 ; xx]
%  yy = [0.5 ; yy]
plot(xx,yy,'*-')
grid on
axis equal
axis square
```

First Cae Study in Polar Coordinate

```
%    Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%    Developed MATLAB Program for
Streamline Simulation
%    This Code developed originally by
MARJAN HASHEM

%    First Case Study in Polar
Coordinate
%    Main route
clc;
clear all;
close all;
format long e
global radiusp T r
N=10;
M=60;
re=1.10;
rw=0.2;
n=(re/rw);
r(1)=rw;
Mu=(10^(-3));
tetha=((2*pi)/M);
Pw=20000;
Pe=100000;

for j=1:M+1
    for i=1:N+1
        if i<=(round(N/2))
    Kr(i,j)=(2*(10^(-12)));
    landar(i,j)=(Kr(i,j)/Mu);
        else
         Kr(i,j)=(2*(10^(-12)));
    landar(i,j)=(Kr(i,j)/Mu);
        end
    end
end
landar


for i=1:N
     m=i/(N);
k=(n)^(m) ;
r(i+1)=rw*k;
end
r
pause
%RP:raduis plus 0.5
for i=1:N

     q=i+0.5;
     p=q-0.5;
RP(p)=((r(i+1)-
r(i))/(log(r(i+1)/r(i))));
end
RP
pause

%RM:raduis mines 0.5
for i=2:N
     C=i-0.5;
     D=C+0.5;
RM(D)=((r(i)-r(i-1))/(log(r(i)/r(i-
1))));
end
RM
pause
for i=1:N
```

```
    q=i+0.5;
    p=q-0.5;
 for   j=1:M

 %landar(i+0.5,j)=landaR(p,j)
landaRP(p,j)=((log(r(i+1)/r(i)))/(((1/l
andar(i,j))*(log(RP(p)/r(i))))+((1/land
ar(i+1,j))*(log(r(i+1)/RP(p))))));
%
landar(p)=((ln(r(i+1)/r(i)))/(((1/landa
r(i))*(ln(r(i+0.5)/r(i))))+((1/landar(i
+1))*(ln(r(i+1)/r(i+0.5))))))
 end
end
landaRP

for i=2:N
    C=i-0.5;
    D=C+0.5;
   for j=1:M

 %landar(i+0.5,j)=landaR(p,j)
landaRM(D,j)=((log(r(i)/r(i-
1)))/(((1/landar(i,j))*(log(r(i)/RM(D))
))+((1/landar(i-1,j))*(log(RM(D)/r(i-
1))))));
%
landar(p)=((ln(r(i+1)/r(i)))/(((1/landa
r(i))*(ln(r(i+0.5)/r(i))))+((1/landar(i
+1))*(ln(r(i+1)/r(i+0.5))))))
   end
end
landaRM
pause



for i=1:N
   for j=1:M
       landat(i,j)=(2*10^(-9));
   end
end
landat
    for i=1:N
        for j=1:M
            u=j+0.5;
            v=u-0.5;
            if j==M

landaT(i,v)=((landat(j)*landat(1))/(lan
dat(j)+landat(1))) ;
            else
landaT(i,v)=((landat(j)*landat(j+1))/(l
andat(j)+landat(j+1)));
%
landat(j+0.5)=((landat(j)*landat(j+1))/
(landat(j)+landat(j+1)))
            end
        end
    end
    landaT



for i=1:N+1
    for j=1:M+1
        landatetha(i,j)=(2*10^(-9));
    end
end
landatetha
```

```
for i=1:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;

landaTETHA(p,j)=(log(r(i+1)/r(i)))/(((1
/landatetha(i,j))*(log(RP(p)/r(i))))+((
1/landatetha(i+1,j))*(log(r(i+1)/RP(p))
)));
%
landatetha(i+0.5)=(ln(r(i+1)/r(i)))/(((
1/landatetha(i))*(ln(r(i+0.5)/r(i))))+(
(1/landatetha(i+1))*(ln(r(i+1)/r(i+0.5)
))))
    end
end
landaTETHA


for i=1:N
    for j=1:M
       u=j+0.5;
       v=u-0.5;
        if j==M

landaTETHAj(i,v)=((landatetha(j)*landat
etha(1))/(landatetha(j)+landatetha(1)))
;
        else
landaTETHAj(i,v)=((landatetha(j)*landat
etha(j+1))/(landatetha(j)+landatetha(j+
1)));
        end
    end
end
landaTETHAj

pause


for i=2:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
         C=i-0.5;
         D=C+0.5;
         RP(p)
             RM(D)
             RP(p)-RM(D)
         if j==M

           a(i,j)=-
(((((tetha)^2)*r(i))/(RP(p)-
RM(D)))*(((landaRP(p,j))/log(r(i+1)/r(i
)))+(landaRM(D,j)/log(r(i)/r(i-1)))))-
(((landat(j)*landat(1))/(landat(j)+land
at(1)))+((landat(j)*landat(j-
1))/(landat(j)+landat(j-1))))
%   pause
         elseif j==1
             a(i,j)=-
(((((tetha)^2)*r(i))/(RP(p)-
RM(D)))*(((landaRP(p,j))/log(r(i+1)/r(i
)))+(landaRM(D,j)/log(r(i)/r(i-1)))))-
(((landat(j)*landat(j+1))/(landat(j)+la
ndat(j+1)))+((landat(j)*landat(M))/(lan
dat(j)+landat(M))))
%          pause
         else
```

```
a(i,j)=-((((tetha)^2)*r(i))/(RP(p)-
RM(D)))*(((landaRP(p,j))/log(r(i+1)/r(i
)))+(landaRM(D,j)/log(r(i)/r(i-1)))))-
(((landat(j)*landat(j+1))/(landat(j)+la
ndat(j+1)))+((landat(j)*landat(j-
1))/(landat(j)+landat(j-1))))
%       pause
        end

    end
end
a
pause
for i=1:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
        C=i-0.5;
        D=C+0.5;
        if j==M

b(i,j)=((landat(j)*landat(1))/(landat(j
)+landat(1)));
        else
b(i,j)=((landat(j)*landat(j+1))/(landat
(j)+landat(j+1)));
        end
    end
end
b
pause
for i=1:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
        C=i-0.5;
        D=C+0.5;
        if j==1

c(i,j)=((landat(j)*landat(M))/(landat(j
)+landat(M)));
        else

c(i,j)=((landat(j)*landat(j-
1))/(landat(j)+landat(j-1)));
        end
    end
end
c
pause

for i=2:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
        C=i-0.5;
        D=C+0.5;
d(i,j)=(((tetha)^2)*r(i))/(RP(p)-
RM(D)))*((landaRM(D,j))/(log(r(i)/(r(i-
1)))));
    end
end
d
pause

for i=2:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
        C=i-0.5;
        D=C+0.5;
e(i,j)=(((tetha)^2)*r(i))/(RP(p)-
RM(D)))*((landaRP(p,j))/(log(r(i+1)/(r(
i)))));
%  e(i,j)=((tetha)^2)*((r(i)^2))/(RP(p)-
RM(D)))*(landaRP(p,j)/(log(r(i+1)/r(i)
))))+(((tetha)*r(i)/2)*((1/(r(i+1)-
r(i)))-(1/(r(i)-r(i-
1))))*(landaTETHA(p,j)));
    end

end
e
pause
% a(i,j)=-
((((tetha)^2)*(r(i)^2))/(R(p)-r(i-
0.5)))*((landar(i+0.5)/ln(r(i+1)/r(i)))
+(landar(i-0.5)/ln(r(i-1)/r(i))))-
(((landat(j)*landat(j+1))/(landat(j)+la
ndat(j+1)))+((landat(j)*landat(j-
1))/(landat(j)+landat(j-1))))
%
b(i,j)=((landat(j)*landat(j+1))/(landat
(j)+landat(j+1)))
% c(i,j)=((landat(j)*landat(j-
1))/(landat(j)+landat(j-1)))
%
d(i,j)=((((tetha)^2)*(r(i)^2))/(r(i+0.5
)-r(i-0.5)))*(landat(i-0.5)/(ln(r(i-
0.5)/r(i))))
%
e(i,j)=((tetha)^2)*(r(i)^2))/(r(i+0.5)-
r(i-
0.5)))*(landat(i+0.5)/(ln(r(i+0.5)/r(i
))))+(((tetha^2)*r(i)/2)*((1/(r(i+0.5)
-r(i-0.5)))-(1/(r(i)-r(i-
1))))*(landat(i+0.5)))

% my alternative when in last series of
equation p1,j-1 change to pN,j-1
% A=zeros((N-1)*M,(N-1)*M)

Z=zeros((N-1)*M,(N-1)*M);
i=2
for j=1:M
    r1=j;
    s1=j;
    Z(r1,s1)=a(i,j);

if j==M
 r2=j;
s2=1;
Z(r2,s2)=b(i,j);
else
 r2=j;
s2=j+1;
Z(r2,s2)=b(i,j);
end

if j==1
r3=j;
s3=M;
Z(r3,s3)=c(i,j);
else
```

```
  r3=j;
  s3=j-1 ;
Z(r3,s3)=c(i,j);
end

    r4=j;
    s4=M*(i-1)+j;
    Z(r4,s4)=e(i,j);


end

for i=3:N-1
    for j=1:M
        r1=M*(i-2)+j;
        s1=j+M*(i-2);
        Z(r1,s1)=a(i,j);

        if j==M
          r2=M*(i-2)+j;
         s2=(1)+M*(i-2);
         Z(r2,s2)=b(i,j) ;
        else
         r2=M*(i-2)+j;
         s2=(j+1)+M*(i-2);
         Z(r2,s2)=b(i,j);
        end

        if j==1
            r3=M*(i-2)+j;
         s3=M*(i-2)+(M);
         Z(r3,s3)=c(i,j) ;
        else
         r3=M*(i-2)+j;
         s3=M*(i-2)+(j-1);
         Z(r3,s3)=c(i,j);
        end

         r4=M*(i-2)+j;
         s4=M*(i-3)+j;
         Z(r4,s4)=d(i,j);

         r5=M*(i-2)+j;
         s5=M*(i-1)+j;
         Z(r5,s5)=e(i,j);
    end
end


 i=N
    for j=1:M
        r1=M*(i-2)+j;
        s1=M*(i-2)+j;
        Z(r1,s1)=a(i,j);

        if j==M

        r2=M*(i-2)+j;
        s2=M*(i-2)+(1);
        Z(r2,s2)=b(i,j);
        else
      r2=M*(i-2)+j;
        s2=M*(i-2)+(j+1);
        Z(r2,s2)=b(i,j);
    end

        if j==1

        r3=M*(i-2)+j;
        s3=M*(i-2)+(M);
        Z(r3,s3)=c(i,j);
        else
        r3=M*(i-2)+j;
        s3=M*(i-2)+(j-1);
        Z(r3,s3)=c(i,j);
        end

        r4=M*(i-2)+j;
        s4=M*(i-3)+j;
        Z(r4,s4)=d(i,j);


    end


 Z  ;



V=zeros(((N-1)*M),1);

i=2
for j=1:M
r1=j;
s1=1;
V(r1,s1)=-(d(i,j)*Pw);
end

 i=N
for j=1:M
    r1=((N-2)*M)+j;
    s1=1;
    V(r1,s1)=-(e(i,j)*Pe);
end
V

P=zeros((N-1)*M ,1);
% (V*inv(A))=P
Q=(Z\V)
P=Q/Pw


k =zeros(N,M);

for i=1:(N-1)
    for j=1:M

        if i==1
            i;
            k(i,j)=Q(j,1);
        else
            i;
            j;
        k(i,j)=Q((((i-1)*M)+j),1);
        end
    end
end
k
for i=N
    for j=1:M
      k(i,j)=Pe;
    end
end

 contourc (k,30) ;
   contour (k,30, 'DisplayName',
'PRESSURE');colormap autumn;
pause
% uRP(i,j)=landaRP(p,j)*(P(i+1,j)-
P(i,j))/(r(i+1)-r(i))

%  ur(p,j)=(landaRP(p,j)*(p(i+1,j)-
p(i,j))/(r(i+1)-
```

```
r(i))+(1/2*RP(p))*((landaTETHAj(i,v)*(p
(i,j+1)-p(i,j))/deltatetha)+(landa



for i=1:N-1
    q=i+0.5;
    p=q-0.5;

    for j=1:M

        if i==N
            1357;
            i;
            j;
            1.5;
            landaRP(p,j);
            2.5;
            k(i,j);
            3.5;
            Pe;
            3.5;
            (Pe-k(i,j));
            3.75;
            r(i+1);
            4.5;
            r(i);
            5.5;
        urp(p,j)=(landaRP(p,j)*(Pe-
k(i,j))/(r(i+1)-r(i)));
        else
            1976;
        i;
        j ;

urp(p,j)=(landaRP(p,j)*(k(i+1,j)-
k(i,j))/(r(i+1)-r(i)));
    end

    end
end
urp



for i=2:N
    for j=1:M
    C=i-0.5;
    D=C+0.5;

  urD(D,j)=(landaRM(D,j)*(k(i,j)-k(i-
1,j))/(r(i)-r(i-1)));
    end
end
urD
for i=1:N
    for j=1:M
        u=j+0.5;
        v=u-0.5;
if j==M


utethav(i,v)=(1/r(i))*(landaT(i,v)*(k(i
,1)-k(i,j)/tetha));
else


utethav(i,v)=(1/r(i))*(landaT(i,v)*(k(i
,j+1)-k(i,j)/tetha));
end

    end
end
utethav

for i=1:N
    for j=1:M
        u=j-0.5;
        w=u+0.5;
        if j==1

utethaw(i,w)=(1/r(i))*(landaT(i,w)*(k(i
,j)-k(i,M)/tetha));
            else


utethaw(i,w)=(1/r(i))*(landaT(i,w)*(k(i
,j)-k(i,j-1)/tetha));
            end
    end
end
utethaw


for i=1:N-1
    q=i+0.5;
    p=q-0.5;

    for j=1:M
        URP(p,1)=urp(p,j);
    end
end
URP
for i=1:N-1
    RPP(i)=RP(i);
end
 RPP

plot(RPP,URP)

% for i=1:N
%   r=[ 2.000000000000000e-001
3.062814313608786e-001
4.690415759823430e-001]
tetha=((2*pi)/M)
for i=1:M
    t(i)=(((2*pi)*(i))/M);
end
t
    T=[0 t]
    K=zeros(N,M+1);
    for i=1:N
        for j=1:M
            K(i,j)=k(i,j);
        end
    end

    for i=1:N

        K(i,M+1) =k(i,1);
    end
k

    K
% teta=[0 45 90 135 180 225 270 360]
for i=1:N
    for j=1:M+1
x(i,j)=r(i)*cos(T(j));
y(i,j)=r(i)*sin(T(j));
    end
end
```

```
x
y
    contourf(x,y,K,50),colormap autumn
    for j=1:M
    urp(N,j)=0.000015;
    end

    for j=1:M
    for i=1:(N-1)
            r(i+1);
            R2=(1/(r(i+1)));
            r(i);
            R1=(1/(r(i)));
            urp(i+1,j);
            qw=urp(i,j);
            ac=((urp(i+1,j)-
urp(i,j))/((R2)-(R1)));
            cd=(R1*(ac));
            bc=((qw)-(cd));
            1/(2*(ac));
            (((r(i+1))^2)-(r(i)^2));

Tr(i)=(1/(2*(ac)))*(((r(i+1))^2)-
(r(i)^2));
             i
             rexit1(i)=Tr(i)*urp(i,j);

rexit2(i)=Tr(i)*urp(i+1,j);
            rexit3(i)=r(i+1)-r(i);

rexit4(i)=((r(i+1)+r(i))/2);
    end

    end
  Tr

%  radiusN=1.10
%  theta(1)=0
   for j=1:M
%    for i=N:1
%            radius(i) =radius(i)-
urp(i)*Tr(i)
 radiusp(N)= 1.10
 re=1.10
 thetap(N,j)= T(j)
   radiuspp=radiusp(N);
   thetapp=thetap(N,j) ;
%    Vtethapa(1,1)=utetha(N,1);
%    urp(1,1)=urp(N,j)
  i=N
%    j=1

 while (radiuspp > rw )&
(thetapp<=(2*pi)) & (i>=2)
    123
%     while ((rppa<re))

radiusp(i-1)=radiusp(i)-(urp(i,j)*Tr(i-
1));
i=i-1
j=j
 end
% radius
plotfinal
 radiuspp=zeros(N,M);
 thetapp=zeros(N,M);
   end
figure
pause
for i=1:N
    for j=1:M
plotfinal2
    end
end
for i=1:N-1
    (RP(i+1)+RP(i))/2
end
% %       end
% %       Tm(i,j)=min
(Ttetha(i,j),Tt(i,j));
% %
Re(i,j)=((1/Atetha(i,j))*((Vtethap(i,j)
*etethap(Atetha(i,j)* Tm(i,j)))-
Vtetha(i,j)))+r(j);
% %
Te(i,j)=((1/At(i,j))*((Vtp(i,j)*etethap
(At(i,j)* Tm(i,j)))-Vt(i,j)))+t(i);
% %      Re(i,j);
% %      Te(i,j);

%%   Streamline Simulation Near Well
Bore
%         By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   First Case Study in Polar
Coordinate
%   Subroutine for drawing the
streamline

global radiusp T
% temp = find(Xp~=0);

x(i,j)=radiusp(i)*cos(T(j))
y(i,j)=radiusp(i)*sin(T(j))
  xx = [ x];
  yy = [ y];
plot(xx,yy,'*-')
 grid on
 axis equal
 axis square

% %%   Streamline Simulation Near Well
Bore
%         By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   First Case Study in Polar
Coordinate
%   Subroutine for drawing the
streamline

global radiusp T r

x(i,j)=r(i)*cos(T(j))
y(i,j)=r(i)*sin(T(j))
  xx = [ x];
```

```
  yy = [ y];
plot(xx,yy,'*-')
 grid on
 axis equal
 axis square
```

Second Cae Study in Polar Coordinate

```
%   Streamline Simulation Near Well
Bore
%          By MARJAN HASHEM

%   Developed MATLAB Program for
Streamline Simulation
%   This Code developed originally by
MARJAN HASHEM

%   Second Case Study in Polar
Coordinate
%   Main route

clc;
clear all;
close all;
format long e
global radiusp TETHA N M i j T
global  urp TMIN utethav RP urpp
utethavpp
global radiuse TETHAe TETHAp
global TETHA  utethav TTETHA
global radiusp TETHA
global  urp TMIN utethav N M i j x y
global radiuse TETHAe TETHAp r utethav
deltatethaexit  checkthesign
global radiusp TETHA Tr1 Tr TTETHA
TTETHA1 radiuse TETHAe tetha
tethamoredniaz checkthesize



N=60;
re=100;
rw=0.2;
n=(re/rw);
M=60;




n=(re/rw);
for j=1:M
r(1,j)=rw;
end;
Mu=(10^(-3));
tetha=((2*pi)/M);
Pw=(0.5*10000000);
Pe=10000000;

Kr=zeros(N+1,M+1);
1
for j=1:M+1
    for i=1:N+1
        if i<=(round(N/2))
   Kr(i,j)=(0.5*(10^(-12)));
   landar(i,j)=(Kr(i,j)/Mu);
        else
         Kr(i,j)=(0.5*(10^(-12)));
    landar(i,j)=(Kr(i,j)/Mu);
        end
    end
end
2
 for j=fix((3*M)/10):fix((7*M)/10)
for i=fix((5*N)/10):fix((6*N)/10)

            Kr(i,j)=(0.5*(10^-17));
            landar(i,j)=(Kr(i,j)/Mu);
        end
 end
      Kr;
landar;

3

for i=1:N
    for j=1:M+1
    m=i/(N);
k=(n)^(m) ;
r(i+1,j)=rw*k;
    end
end
r;
4
for i=1:N
    for j=1:M
    q=i+0.5;
    p=q-0.5;
RP(p,j)=((r(i+1,j)-
r(i,j))/(log(r(i+1,j)/r(i,j))));
    end
end
RP;
5
for i=2:N
    for j=1:M
    C=i-0.5;
    D=C+0.5;
RM(D,j)=((r(i,j)-r(i-
1,j))/(log(r(i,j)/r(i-1,j))));
    end
end
RM;
6
% pause
for i=1:N
    q=i+0.5;
    p=q-0.5;
 for   j=1:M

landaRP(p,j)=((log(r(i+1,j)/r(i,j)))/((
(1/landar(i,j))*(log(RP(p,j)/r(i,j))))+
((1/landar(i+1,j))*(log(r(i+1,j)/RP(p,j
)))))));
end
end
landaRP;
7
for i=2:N
    C=i-0.5;
    D=C+0.5;
   for j=1:M

landaRM(D,j)=((log(r(i,j)/r(i-
1,j)))/((((1/landar(i,j))*(log(r(i,j)/RM
(D,j))))+((1/landar(i-
1,j))*(log(RM(D,j)/r(i-1,j))))));
   end
end
landaRM;
8
```

```
for i=1:N
   for j=1:M
       landat(i,j)=(2*10^(-9));
   end
end
landat;
9
    for i=1:N
        for j=1:M
            u=j+0.5;
            v=u-0.5;
            if j==M

landaTp(i,j)=((landat(i,j)*landat(i,1))
/(landat(i,j)+landat(i,1))) ;

            else
landaTp(i,j)=((landat(i,j)*landat(i,j+1
))/(landat(i,j)+landat(i,j+1)));
            end
        end
    end
    landaTp;
    10

        for i=1:N
        for j=1:M
            if j==1

landaTd(i,j)=((landat(i,j)*landat(i,M))
/(landat(i,j)+landat(i,M))) ;

            else
landaTd(i,j)=((landat(i,j)*landat(i,j-
1))/(landat(i,j)+landat(i,j-1)));
%
landat(j+0.5)=((landat(j)*landat(j+1))/
(landat(j)+landat(j+1)))
            end
        end
    end
    landaTd;
    10.5




for i=1:N+1
    for j=1:M+1
        landatetha(i,j)=(2*10^(-9));
    end
end
landatetha;
11
for i=1:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;

landaTETHA(p,j)=(log(r(i+1,j)/r(i,j)))/
(((1/landatetha(i,j))*(log(RP(p,j)/r(i,
j))))+((1/landatetha(i+1,j))*(log(r(i+1
,j)/RP(p,j)))));
    end
end
landaTETHA;

12
for i=1:N
    for j=1:M
       u=j+0.5;
       v=u-0.5;
        if j==M

landaTETHAj(i,v)=2*((landatetha(i,j)*la
ndatetha(i,1))/(landatetha(i,j)+landate
tha(i,1))) ;
        else
landaTETHAj(i,v)=2*((landatetha(i,j)*la
ndatetha(i,j+1))/(landatetha(i,j)+landa
tetha(i,j+1)));
        end
    end
end
landaTETHAj;
13
% pause


for i=2:N
    for j=1:M
        q=i+0.5;
        p=q-0.5;
         C=i-0.5;
         D=C+0.5;
         RP(p,j);
             RM(D,j);
             RP(p,j)-RM(D,j);
         if j==M

           a(i,j)=-
(((((tetha)^2)*r(i,j))/(RP(p,j)-
RM(D,j)))*(((landaRP(p,j))/log(r(i+1,j)
/r(i,j)))+(landaRM(D,j)/log(r(i,j)/r(i-
1,j)))))-
(((landat(i,j)*landat(i,1))/(landat(i,j
)+landat(i,1)))+((landat(i,j)*landat(i,
j-1))/(landat(i,j)+landat(i,j-1))));
%   pause
         elseif j==1
             a(i,j)=-
(((((tetha)^2)*r(i,j))/(RP(p,j)-
RM(D,j)))*(((landaRP(p,j))/log(r(i+1,j)
/r(i,j)))+(landaRM(D,j)/log(r(i,j)/r(i-
1,j)))))-
(((landat(i,j)*landat(i,j+1))/(landat(i
,j)+landat(i,j+1)))+((landat(i,j)*landa
t(i,M))/(landat(i,j)+landat(i,M))));
%         pause
         else

a(i,j)=-
(((((tetha)^2)*r(i,j))/(RP(p,j)-
RM(D,j)))*(((landaRP(p,j))/log(r(i+1,j)
/r(i,j)))+(landaRM(D,j)/log(r(i,j)/r(i-
1,j)))))-
(((landat(i,j)*landat(i,j+1))/(landat(i
,j)+landat(i,j+1)))+((landat(i,j)*landa
t(i,j-1))/(landat(i,j)+landat(i,j-
1))));
%         pause
         end

    end
end
a;
14
% pause
for i=1:N
    for j=1:M
```

```
            q=i+0.5;
            p=q-0.5;
             C=i-0.5;
             D=C+0.5;
             if j==M

b(i,j)=((landat(i,j)*landat(i,1))/(land
at(i,j)+landat(i,1)));
             else
b(i,j)=((landat(i,j)*landat(i,j+1))/(la
ndat(i,j)+landat(i,j+1)));
             end
      end
end
b;
15
% pause
for i=1:N
      for j=1:M
            q=i+0.5;
            p=q-0.5;
             C=i-0.5;
             D=C+0.5;
             if j==1

c(i,j)=((landat(i,j)*landat(i,M))/(land
at(i,j)+landat(i,M)));
             else

c(i,j)=((landat(i,j)*landat(i,j-
1))/(landat(i,j)+landat(i,j-1)));
             end
      end
end
c;
16
% pause

for i=2:N
      for j=1:M
            q=i+0.5;
            p=q-0.5;
             C=i-0.5;
             D=C+0.5;
d(i,j)=((((tetha)^2)*r(i,j))/(RP(p,j)-
RM(D,j)))*((landaRM(D,j))/(log(r(i,j)/(
r(i-1,j)))));
      end
end
d;
17
% pause

for i=2:N
      for j=1:M
            q=i+0.5;
            p=q-0.5;
             C=i-0.5;
             D=C+0.5;
e(i,j)=((((tetha)^2)*r(i,j))/(RP(p,j)-
RM(D,j)))*((landaRP(p,j))/(log(r(i+1,j)
/(r(i,j)))));
      end

end
e;
18

Z=zeros((N-1)*M,(N-1)*M);
i=2
for j=1:M
             r1=j;
             s1=j;
             Z(r1,s1)=a(i,j);

      if j==M
       r2=j;
      s2=1;
      Z(r2,s2)=b(i,j);
      else
       r2=j;
      s2=j+1;
      Z(r2,s2)=b(i,j);
      end

      if j==1
      r3=j;
      s3=M;
      Z(r3,s3)=c(i,j);
      else
        r3=j;
        s3=j-1 ;
      Z(r3,s3)=c(i,j);
      end

             r4=j;
             s4=M*(i-1)+j;
             Z(r4,s4)=e(i,j);


end
19
for i=3:N-1
      for j=1:M
            r1=M*(i-2)+j;
             s1=j+M*(i-2);
             Z(r1,s1)=a(i,j);

             if j==M
               r2=M*(i-2)+j;
              s2=(1)+M*(i-2);
              Z(r2,s2)=b(i,j) ;
             else
              r2=M*(i-2)+j;
              s2=(j+1)+M*(i-2);
              Z(r2,s2)=b(i,j);
             end

             if j==1
                 r3=M*(i-2)+j;
              s3=M*(i-2)+(M);
              Z(r3,s3)=c(i,j) ;
             else
              r3=M*(i-2)+j;
              s3=M*(i-2)+(j-1);
              Z(r3,s3)=c(i,j);
             end

              r4=M*(i-2)+j;
              s4=M*(i-3)+j;
              Z(r4,s4)=d(i,j);

              r5=M*(i-2)+j;
              s5=M*(i-1)+j;
              Z(r5,s5)=e(i,j);
      end
end
20

 i=N;
    for j=1:M
         r1=M*(i-2)+j;
```

```
        s1=M*(i-2)+j;
        Z(r1,s1)=a(i,j);

        if j==M

        r2=M*(i-2)+j;
        s2=M*(i-2)+(1);
        Z(r2,s2)=b(i,j);
        else
      r2=M*(i-2)+j;
        s2=M*(i-2)+(j+1);
        Z(r2,s2)=b(i,j);
    end

        if j==1

         r3=M*(i-2)+j;
         s3=M*(i-2)+(M);
         Z(r3,s3)=c(i,j);
        else
         r3=M*(i-2)+j;
         s3=M*(i-2)+(j-1);
         Z(r3,s3)=c(i,j);
        end

         r4=M*(i-2)+j;
         s4=M*(i-3)+j;
         Z(r4,s4)=d(i,j);


    end

 Z  ;

21

V=zeros(((N-1)*M),1);
22
i=2
for j=1:M
r1=j;
s1=1;
V(r1,s1)=-(d(i,j)*Pw);
end

23
i=N
for j=1:M
    r1=((N-2)*M)+j;
    s1=1;
    V(r1,s1)=-(e(i,j)*Pe);
end
V;
24
P=zeros((N-1)*M ,1);
% (V*inv(A))=P
Q=(Z\V);
P=Q/Pw;
25

k =zeros(N,M);

for i=1:(N-1)
    for j=1:M

        if i==1
            i;
          k(i,j)=Q(j,1);
        else
            i;
            j;
            k(i,j)=Q((((i-1)*M)+j),1);
            end
        end
end
k ;
26
for i=N
    for j=1:M
        k(i,j)=Pe;
    end
end
27

contourf (k,30, 'DisplayName',
'PRESSURE');colormap autumn;

for i=1:N-1
    q=i+0.5;
    p=q-0.5;

    for j=1:M

        if i==(N)
          1357;
           i;
           j;
           1.5;
          landaRP(p,j);
          2.5;
          k(i,j);
          3.5;
          Pe;
          3.5;
          (Pe-k(i,j));
          3.75;
          r(i+1);
          4.5;
          r(i);
          5.5;
        urp(p,j)=(landaRP(p,j)*((Pe-
k(i,j))/(r(i+1,j)-r(i,j))));
        else
           1976;
        i;
      j ;

urp(p,j)=(landaRP(p,j)*((k(i+1,j)-
k(i,j))/(r(i+1,j)-r(i,j))));
    end

    end
end
urp;
28


for i=2:N
    for j=1:M
    C=i-0.5;
    D=C+0.5;

  urD(D,j)=(landaRM(D,j)*((k(i,j)-k(i-
1,j))/(r(i,j)-r(i-1,j))));
    end
end
urD;
29
for i=1:N
    for j=1:M
        u=j+0.5;
         v=u-0.5;
```

```
if j==M

utethav(i,v)=(1/r(i,j))*(landaTp(i,j)*(
(k(i,1)-k(i,j))/(tetha)));
else

utethav(i,v)=(1/r(i,j))*(landaTp(i,j)*(
(k(i,j+1)-k(i,j))/(tetha)));
end

        end
end
utethav;
30
for i=1:N
        for j=1:M
                u=j-0.5;
                w=u+0.5;
                if j==1

utethaw(i,w)=(1/r(i,j))*(landaTd(i,j)*(
(k(i,j)-k(i,M))/(tetha)));
                else


utethaw(i,w)=(1/r(i,j))*(landaTd(i,j)*(
(k(i,j)-k(i,j-1))/(tetha)));
                end
        end
end
utethaw;
31
% URP
for i=1:N-1
   for j=1:M
      RPP(i,j)=RP(i,j);
end
end
   RPP  ;
32
% pause
tetha=((2*pi)/M)
for i=1:N
        for j=1:M
           t(i,j)=(((2*pi)*(j))/M);
        end
end
t;

33
for i=1:N
        for j=1
           T(i,j)=0;
        end
end
for i=1:N
        for j=2:M+1
           T(i,j)=t(i,j-1);
        end
end
% T=
% T=[0 t]
34
        K=zeros(N,M+1);
        for i=1:N
           for j=1:M
              K(i,j)=k(i,j);
           end
        end
```

```
        for i=1:N

           K(i,M+1) =k(i,1);
        end
k;
   K;
   35
for i=1:N
   for j=1:M+1
f(i,j)=r(i,j)*cos(T(i,j));
g(i,j)=r(i,j)*sin(T(i,j));
   end
end
 f;
 g;
35.5
contourf(f,g,K,20),colormap autumn
        figure
36
37
   for j=1:M
        for i=N
   radiusp(i,j)=RP(i,j);
        end
   end

38

 i=N-1;
 j=1;
 1970
%   j=1
   TETHAp(i,1)= 0;
   radiuspp=radiusp(i,1);
   39
   TETHApp=TETHAp(i,1) ;
   utethavpp(i,j)=utethav(i,j);
%    pause;
   urpp(i,1)=urp(i,j);
    while (radiusp(i,j)>(RP(2,1)))
    1971
locationnnew
12345
radiuse;
TETHAe;


if TMIN(i,j)==Tr(i,j)
        1983

radiusp(i-1,j)=radiuse(i,j);
TETHAp(i-1,j)=TETHAe(i,j);
TETHApp=TETHAp(i-1,j);
radiuspp=radiusp(i-1,j);
i=i-1;
j=j;

        elseif TMIN(i,j)==TTETHA(i,j)

            if deltatethaexit(i,j)<0

        if j==1
radiusp(i,M)=radiuse(i,j);
TETHAp(i,M)=TETHAe(i,j);
TETHApp=TETHAp(i,M);
radiuspp=radiusp(i,M) ;
           j=M;
           i=i;
        else
radiusp(i,j-1)=radiuse(i,j);
```

```
TETHAp(i,j-1)=TETHAe(i,j);
TETHApp=TETHAp(i,j-1);
radiuspp=radiusp(i,j-1) ;
            j=j-1;
            i=i;
        end
        1984

    else
        if j==M
            1985

 radiusp(i,1)=radiuse(i,j);
TETHAp(i,1)=TETHAe(i,j);
TETHApp=TETHAp(i,1);
radiuspp=radiusp(i,1) ;
            j=1;
            i=i;


        else
            1986
 radiusp(i,j+1)=radiuse(i,j);
TETHAp(i,j+1)=TETHAe(i,j);
TETHApp=TETHAp(i,j+1);
radiuspp=radiusp(i,j+1) ;
            j=j+1;
            i=i;

        end
    else
        fprint('nemidoonam')
 end
end
 2000
    urpp=zeros(N-1,M);
   utethavpp=zeros(N-1,M);
   radiusp=zeros(N-1,M);
   TETHAp=zeros(N-1,M);
   radiuspp=zeros(N-1,M);
   TETHApp=zeros(N-1,M);
1980
  for j=2:M
    i=N-1
    j;
    i;
  radiusp(i,j)=RP(i,j);
radiuspp=radiusp(i,j);

   TETHAp(i,j)=T(i,j);
   TETHApp=TETHAp(i,j)    ;
  utethavpp(i,j)=utethav(i,j);
   urpp(i,j)=urp(i,j);
   while
((radiusp(i,j)>RP(2,1))&(TETHApp<=(2*pi
)))
    1981

locationnnew
radiuse;
TETHAe;
1982

    if TMIN(i,j)==Tr(i,j)
       1983

radiusp(i-1,j)=radiuse(i,j);
TETHAp(i-1,j)=TETHAe(i,j);
TETHApp=TETHAp(i-1,j);
radiuspp=radiusp(i-1,j);
i=i-1;
j=j;

        elseif TMIN(i,j)==TTETHA(i,j)

          if deltatethaexit(i,j)<0

          if j==1
radiusp(i,M)=radiuse(i,j);
TETHAp(i,M)=TETHAe(i,j);
TETHApp=TETHAp(i,M);
radiuspp=radiusp(i,M) ;
            j=M;
            i=i;
        else
radiusp(i,j-1)=radiuse(i,j);
TETHAp(i,j-1)=TETHAe(i,j);
TETHApp=TETHAp(i,j-1);
radiuspp=radiusp(i,j-1) ;
            j=j-1;
            i=i;
        end
        1984

    else
        if j==M
            1985

 radiusp(i,1)=radiuse(i,j);
TETHAp(i,1)=TETHAe(i,j);
TETHApp=TETHAp(i,1);
radiuspp=radiusp(i,1) ;
            j=1;
            i=i;


        else
            1986
 radiusp(i,j+1)=radiuse(i,j);
TETHAp(i,j+1)=TETHAe(i,j);
TETHApp=TETHAp(i,j+1);
radiuspp=radiusp(i,j+1) ;
            j=j+1;
            i=i;

        end
    end
    else
        fprint('nemidoonam')
 end
    end
 radiusp=zeros(N-1,M);
   TETHAp=zeros(N-1,M);
   radiuspp=zeros(N-1,M);
   TETHApp=zeros(N-1,M);


 1388
 3010

 end

global radiuse TETHAe

    for j=1:M
      for  i=1:N-1
           j;
           i;

hash(i,j)=radiuse(i,j)*cos(TETH
Ae(i,j));
```

```
bash(i,j)=radiuse(i,j)*sin(TETH
Ae(i,j));

          plot(hash,bash,'-')
          grid on
 axis equal
 axis square

      end
    end
```