Partial Differential Equations

A. Godunov

1. Parabolic PDE
2. Diffusion equation: explicit methods
3. Diffusion equation: implicit methods
4. Convection-Diffusion equation

updated 7 April 2022

1

Part 1:

Parabolic PDEs

2

Parabolic PDEs

The diffusion equation

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial^2 x}$$

The diffusion-convection equation

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial^2 x} - b\frac{\partial f}{\partial x}$$

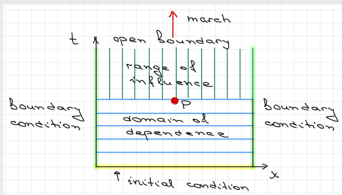
where a is the diffusivity coefficient, and b is the convection velocity.

Parabolic PDEs are are initial-boundary-value problems in open domains (open with respect to time or a time-like variable) in which the solution in the domain of interest is marched forward from the initial state, guided and modified by the boundary conditions.

3

3

The domain of the solution and boundary conditions

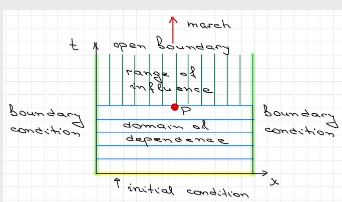


The parabolic PDEs have an infinite physical information propagation speed. As a result, the solution at a given point P at time level n depends on the solution at all other points in the solution domain at all times preceding and including time level n, and the solution at a given point P at time level n influences the solution at all other points in the solution domain at all times including and after time level n.

4

4

The boundary conditions



The solution must satisfy an initial condition at $t = 0, f(x, 0) = F(x)$. The time coordinate has an unspecified (i.e., open) final value.

Since parabolic PDEs above are second order in the spatial coordinate, two boundary conditions are required. These may be of the Dirichlet type (as $f_1(0, t), f_2(L, t)$), the Neumann type, or the mixed type.

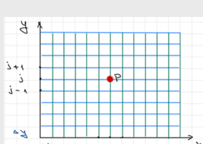
5

5

Finite-difference method

The finite difference method is a numerical procedure which solves a partial differential equation (PDE) by

1. discretizing the continuous physical domain into a discrete finite difference grid,
2. approximating the individual exact partial derivatives in the PDE by algebraic finite difference approximations (FDAs),
3. substituting the FDAs into the PDE to obtain an algebraic finite difference equation (FDE),
4. and solving the resulting algebraic finite difference equations (FDEs)



6

6

Part 2:

Diffusion equation: explicit methods

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial^2 x}$$

7

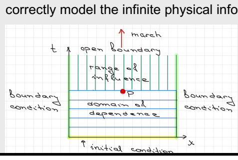
Explicit methods

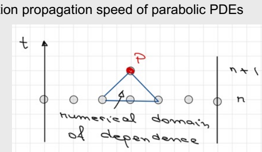
The objective of the numerical solution of a parabolic PDE is to march the solution at time level n forward in time to time level n+1.

In view of the infinite physical information propagation associated with parabolic PDEs, the solution at point P at time level n+1 depends on the solution at all of the other points at time level n+1.

Finite difference methods when the solution at point *P* at time level *n*+1 depends only on the solution at neighboring points at time level *n* are called explicit methods. Explicit methods are computationally faster than implicit methods because there is no system of finite difference equations to solve.

However, the finite numerical information propagation speed of explicit methods does not correctly model the infinite physical information propagation speed of parabolic PDEs





8

8

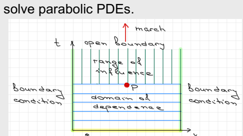
Implicit methods

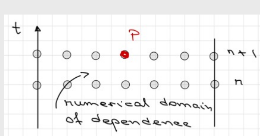
Finite difference methods in which the solution at point *P* at time level n+1 depends on the solution at neighboring points at time level n+1 as well as the solution at time level *n* have an infinite numerical information propagation speed. Such finite difference methods are called implicit methods.

implicit methods appear to be well suited for solving parabolic PDEs, and explicit methods appear to be unsuitable for solving parabolic PDEs.

In actuality, only an infinitesimal amount of physical information propagates at the infinite physical information propagation speed. The bulk of the physical information travels at a finite physical information propagation speed.

Experience has shown that explicit methods as well as implicit methods can be employed to solve parabolic PDEs.



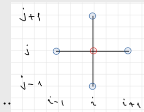


9

9

Finite Difference Approximation

$$f_t = af_{xx}$$



Using Taylor series for the time derivative

$$f_{i,j+1} = f_{i,j+1} + f_t\Big|_{i,j}\Delta t + \frac{1}{2}f_{tt}\Big|_{i,j}\Delta t^2 + \cdots$$

and the first-order forward-time derivative

$$f_t\Big|_{i,j} = \frac{f_{i,j+1} - f_{i,j}}{\Delta t}$$

For the space derivative (same was as for the Laplace equation)

$$f_{xx}\Big|_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

Second-order centered- difference.

10

10

The forward-time centered-space (FTCS) method

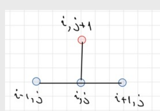
The parabolic PDE $f_t = af_{xx}$

$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = a\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

Solving for $f_{i,j+1}$

$$f_{i,j+1} = f_{i,j} + d(f_{i+1,j} - 2f_{i,j} + f_{i-1,j})$$

where $d = a\Delta t/\Delta x^2$ is called a diffusion number



11

11

Attention

$$f_{i,j+1} = f_{i,j} + d(f_{i+1,j} - 2f_{i,j} + f_{i-1,j})$$

where $d = a\Delta t/\Delta x^2$ is called a diffusion number

Numerical solutions for $d \geq 1$ are numerically unstable

Using von Neuman method* gives that $d \leq 0.5$ (at least) for stability.

12

12

Example: MatLab code

```
%{
  Solving the diffusion equation with Dirichlet BCs
  Method: Forward-time Centered-space difference
  INPUT:
  f(i,j)    boundary conditions
  dx, dy    grid increments
  nx        number of grid points in x direction
  nt        number of grid points in y direction
  a         diffusion coefficient
  OUTPUT
  f2(x,y)   the solution
%}
function[f2] = pdeP1(f,dx,dt,nx,nt,a)
% preparation
f2 = zeros(nx,nt);
d = a*dt/(dx^2);
fprintf(' Diffusion parameter d = %6.4f \n',d)
if d > 0.5
  fprintf(' ATTENTION: the diffusion parameter is too large for FTCS method \n ')
end
for j=1:nt
  for i=2:nx-1
    f(i,j+1) = f(i,j)+d*(f(i+1,j)-2.0*f(i,j)+f(i-1,j));
  end
end
% Prepare OUTPUT
for j=1:nt
  for i=1:nx
    f2(i,j)= f(i,j);
  end
end
end
```

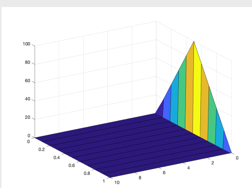
13

13

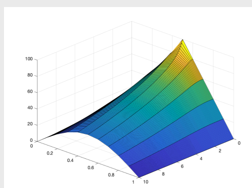
Example: Using FTCS method

$$f_t = a f_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.1$

`Time – less than 1 second`

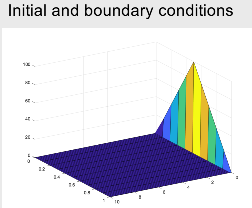
14

14

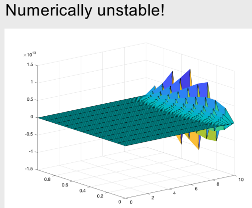
Example: FTCS numerically unstable with d=0.6

$$f_t = a f_{xx}$$

Initial and boundary conditions



Numerically unstable!



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.6$

`Time – less than 1 second`

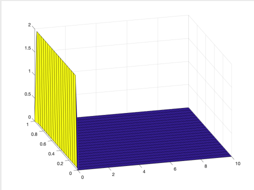
15

15

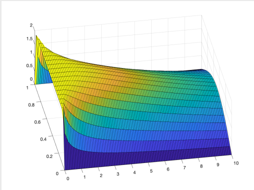
Example: Using FTCS method

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.4$

`Time – less than 1 second`

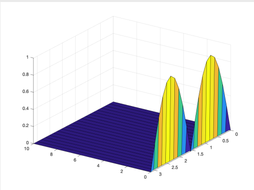
16

16

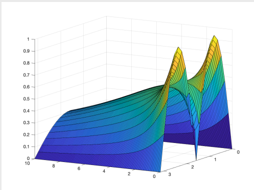
Example: Using FTCS method

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.4$

`Time – less than 1 second`

17

17

Summary for the FTCS

In summary, the forward-time centered-space (FTCS) approximation of the diffusion equation is

- explicit,
- single step,
- consistent,
- $O(\Delta t) + O(\Delta x^2)$,
- conditionally stable,
- and convergent.

It is somewhat inefficient because the time step varies as the square of the spatial grid size.

18

18

The Richardson and Dufort-Frankel methods

The forward-time centered-space(FTCS) approximation of the diffusion equation $f_t = a\, f_{xx}$ has several desirable features.

It is an explicit, two-level, single-step method.

The finite difference approximation of the spatial derivative is second order. However, the finite difference approximation of the time derivative is only first order.

An obvious improvement would be to use a second-order finite difference approximation of the time derivative.

The Richardson (leapfrog) and Dufort-Frankel methods are two such methods.

19

19

The Richardson (Leapfrog) methods

Richardson proposed approximating the partial derivative f_t by the three-level second-order centered-difference approximation based on time levels $j-1, j$, and $j+1$.

$$f_t\Big|_{i,j} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta t}$$

Then the diffusion equation

$$\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta t} = a\,\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

$$f_{i,j+1} = f_{i,j-1} + 2d\left(f_{i+1,j} - 2f_{i,j} + f_{i-1,j}\right), \qquad d = a\Delta t/\Delta x^2$$

The Richardson method appears to be a significant improvement over the FTCS method because of the increased accuracy of the finite difference approximation of f_t. BUT!!! Since the Richardson method is unconditionally unstable when applied to the diffusion equation, it cannot be used to solve that equation, or any other parabolic PDE

20

20

The Dufort-Frankel method

Dufort and Frankel (1953) proposed a modification to the Richardson method for the diffusion equation $f_t = a\, f_{xx}$ which removes the unconditional instability.

$$(1+2d)f_{i,j+1} = (1-2d)f_{i,j-1} + 2d\left(f_{i+1,j} + f_{i-1,j}\right).$$

However this equation is not a consistent approximation of the diffusion equation as $\Delta t \to 0, \Delta x \to 0$.

Due to the inconsistency the Dufort-Frankel method is not an acceptable method for solving the parabolic diffusion equation, or any other parabolic PDE.

21

21

Part 3:

Diffusion equation: implicit methods

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial^2 x}$$

22

Implicit methods: Pros and cons.

In implicit methods, the finite difference approximations of the individual exact partial derivatives in the partial differential equation are evaluated at the solution time level $n+1$.

Implicit difference methods are unconditionally stable. There is no limit on the allowable time step required to achieve a numerically stable solution. There is, of course, some practical limit on the time step required to maintain the truncation errors within reasonable limits, but this is not a stability consideration; it is an accuracy consideration.

Implicit methods do have some disadvantages.

1. The solution at a point in the solution time level $n+1$ depends on the solution at neighboring points in the solution time level, which are also unknown.
2. Additional complexities arise when the partial differential equations are nonlinear.

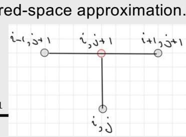
23

23

The Backward-Time Centered-Space (BTCS) Method

The finite difference equation which approximates the partial differential equation is obtained by replacing the exact partial derivative f_t by the first-order backward-time approximation, and the exact partial derivative f_{xx} by the second-order centered-space approximation.

$$f_t\big|_{i,j+1} = \frac{f_{i,j+1} - f_{i,j}}{\Delta t}$$



$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = a\,\frac{f_{i+1,j+1} - 2f_{i,j+1} + f_{i-1,j+1}}{\Delta x^2}$$

$$-df_{i-1,j+1} + (1+2d)f_{i,j+1} - df_{i+1,j+1} = f_{i,j}, \qquad d = a\Delta t/\Delta x^2$$

Equation above cannot be solved explicitly for $f_{i,j+1}$ because the two unknown neighboring values $f_{i-1,j+1}$ and $f_{i+1,j+1}$ also appear in the equation. Thus, we need to solve a tri-diagonal system of linear equations.

24

24

Stability, convergence and consistency.

The BTCS approximation of the diffusion equation is consistent and convergent.

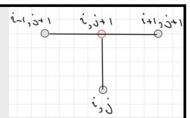
The BTCS method is unconditionally stable. The time step can be much larger than the time step for the FTCS method.

Consequently, the solution at a given time level can be reached with much less computational effort by taking time steps much larger than those allowed for the FTCS method. In fact, the time step is limited only by accuracy requirements.

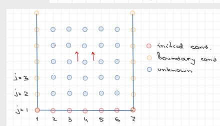
25

25

Application of the BTCS approximation



For Dirichlet boundary conditions



$(1+2d)f_{2,j+1} - df_{3,j+1} = f_{2,j} + df_{1,j+1}$

$-df_{2,j+1} + (1+2d)f_{3,j+1} - df_{4,j+1} = f_{3,j}$

$-df_{3,j+1} + (1+2d)f_{4,j+1} - df_{5,j+1} = f_{4,j}$

… … … … … … … .

$-df_{imax-2,j+1} + (1+2d)f_{imax-1,j+1} = f_{imax-1,j} + df_{imax,j+1}$

The system above comprises a tridiagonal system of linear algebraic equations.

The system can be solved very efficiently by the Thomas algorithm.

Since the coefficient matrix does not change from one time level to the next, LU factorization can be employed with the Thomas algorithm to reduce the computational effort even further.

26

26

Example: MatLab code

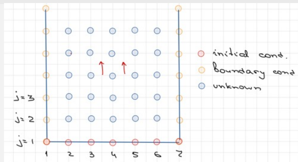
```
%{
  Solving the diffusion equation with Dirichlet BCs
  Method: Backward-time Centered-space difference
  INPUT:
  f(i,j)    initial matrix with boundary conditions
  dx, dy    grid increments
  nx        number of grid points in x direction
  nt        number of grid points in y direction
  alpha     diffusion coefficient
  OUTPUT
  f2(x,y)   the solution
AG: April 2022
%}
function[f2] = pdeP2(f,dx,dt,nx,nt,alpha)
% preparation
a  = zeros(nx-2,3);
b  = zeros(1,nx-2);
f2 = zeros(nx,nt);
d = alpha*dt/(dx^2);
fprintf(' Diffusion parameter d = %6.4f \n',d)
```

27

27

Example: MatLab code

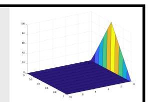
```
for j=2:nt
  for i=2:nx-1
    a(i-1,1) = 0.0-d;
    a(i-1,2) = 1.0+2.0*d;
    a(i-1,3) = 0.0-d;
    b(i-1) = f(i,j-1);
  end
    a(1,1) = 0.0;
    b(1) = b(1) + d*f(1,j);
    a(nx-2,3)= 0.0;
    b(nx-2) = b(nx-2) + d*f(nx,j);

  [w] = Thomas(a,b,nx-2);
  for i=1:nx-2
    f(i+1,j) = w(i);
  end
end
% Prepare OUTPUT
for j=1:nt
  for i=1:nx
    f2(i,j)= f(i,j);
  end
end
end
```



28

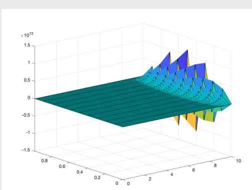
28

Example: comparing FTCS and BTCS

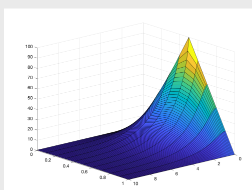


$$f_t = af_{xx}$$

FTCS



BTCS - stable!



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.6$

The BTCS is stable even for any d!

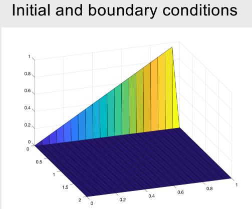
29

29

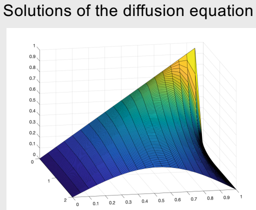
Example: Using BTCS method

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.4$

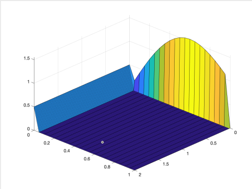
30

30

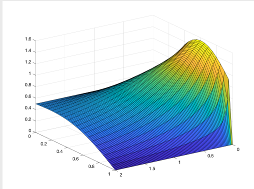
Example: Using BTCS method

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 0.8$

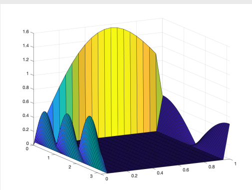
31

31

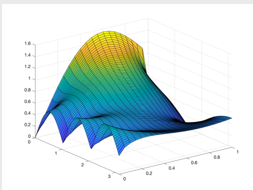
Example: Using BTCS method

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 1.25$

32

32

Nonlinear PDs and multi-dimensional problems

The BTCS method can be used to solve nonlinear PDEs, systems of PDEs, and multidimensional problems.

However, in those cases, the solution procedure becomes quite complicated.

33

33

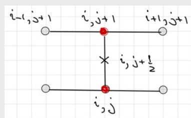
The Crank-Nicolson method

In the backward-time centered-space (BTCS) approximation the spatial derivative is second order. However, the finite difference approximation of the time derivative is only first order.

Using a second-order finite difference approximation of the time derivative would be an obvious improvement. Crank and Nicolson proposed to use the grid point $i, j+\frac{1}{2}$ as the base point in Taylor series for f_t derivative (second-order central difference)

$$f_t\Big|_{i,j+\frac{1}{2}} = \frac{f_{i,j+1} - f_{i,j}}{\Delta t}, \qquad \text{and average for} \quad f_{xx}\Big|_{i,j+\frac{1}{2}} = \frac{1}{2}\left(f_{xx}\Big|_{i,j+1} + f_{xx}\Big|_{i,j}\right)$$

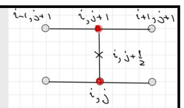
with the central-difference for $f_{xx}|_{i,j+1}$ and $f_{xx}|_{i,j}$



34

34

The Crank-Nicolson method



The diffusion equation then reads

$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = \frac{1}{2}a\left(\frac{f_{i+1,j+1} - 2f_{i,j+1} + f_{i-1,j+1}}{\Delta x^2} + \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}\right)$$

Rearranging terms gives the finite difference equation

$$-df_{i-1,j+1} + 2(1+d)f_{i,j+1} - df_{i+1,j+1} = df_{i-1,j} + 2(1-d)f_{i,j} + df_{i+1,j}$$

where $d = a\Delta t/\Delta x^2$ is a diffusion number

The system of equations can be solved by the Thomas algorithm (three unknowns in a row).

In summary, the Crank-Nicolson approximation of the diffusion equation is implicit, single step, consistent, $O(\Delta t^2) + O(\Delta x^2)$, unconditionally stable, and convergent.

35

35

For Dirichlet boundary conditions

$2(1+d)f_{2,j+1} - df_{3,j+1} = df_{1,j} + 2(1-d)f_{2,j} + df_{3,j} + df_{1,j+1}$

$-df_{2,j+1} + 2(1+d)f_{3,j+1} - df_{4,j+1} = df_{2,j} + 2(1-d)f_{3,j} + df_{4,j}$

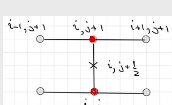
$-df_{3,j+1} + 2(1+d)f_{4,j+1} - df_{5,j+1} = df_{3,j} + 2(1-d)f_{4,j} + df_{5,j}$

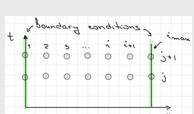
………….

$-df_{N-2,j+1} + 2(1+d)f_{N-1,j+1} = df_{N-2,j} + 2(1-d)f_{N-1,j} + df_{N,j} + df_{N,j+1}$

where $N = i_{max}$

This is a tridiagonal system of linear equations.





36

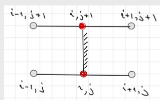
36

Derivative boundary conditions

The implementation of a derivative boundary condition does not depend on whether the problem is an equilibrium problem or a propagation problem

Consequently, the procedure for implementing a derivative boundary condition for one-dimensional equilibrium problems can be applied directly to one-dimensional propagation problems.

$$f_{i+1,j} = f_{i-1,j} + 2\Delta x \, f_x \Big|_{i,j}$$



37

37

More on implicit methods

Extensions:

- The derivative boundary conditions
  see books on numerical methods
  + using ideas similar to elliptical PDEs
- Non-linear equations
  using Newton's method for solving a system of non-linear equations
- Multidimensional problems
  2D problem yields a system based on a banded penta-diagonal matrix
  Successive-overrelaxation will work for explicit methods
  3D problem – use ADI (alternating-direction-implicit) method and AFI (approximate-factorization-implicit) method.

38

38

Asymptotic steady-state solution to propagation problems

Marching methods are employed for solving unsteady propagation problems, which are governed by parabolic and hyperbolic partial differential equations. The emphasis in those problems is on the transient solution itself.

Marching methods also can be used to solve steady equilibrium problems and steady mixed (i.e., elliptic-parabolic or elliptic-hyperbolic) problems as the asymptotic solution in time of an appropriate unsteady propagation problem.

Mixed problems present serious numerical difficulties due to the different types of solution domains (closed domains for equilibrium problems and open domains for propagation problems) and different types of auxiliary conditions (boundary conditions for equilibrium problems and boundary conditions and initial conditions for propagation problems).

Consequently, it may be easier to obtain the solution of a steady mixed problem by reposing the problem as an unsteady parabolic or hyperbolic problem and using marching methods to obtain the asymptotic steady state solution.

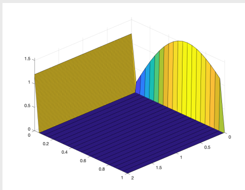
39

39

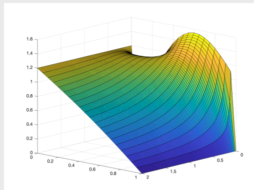
Example: Reaching asymptotic steady state solution

$$f_t = af_{xx}$$

Initial and boundary conditions



Solutions of the diffusion equation



the diffusion parameter $d = a\Delta t/\Delta x^2 = 1.2$

40

40

Part 4:

Convection-Diffusion equation

$$\frac{\partial f}{\partial t} = a\frac{\partial^2 f}{\partial^2 x} - b\frac{\partial f}{\partial x}$$

41

The convection-diffusion equation

1D equation

$$f_t = af_{xx} - bf_x$$

where b is the convection velocity and a is the diffusion coefficient.

The convection-diffusion equation applies to problems in mass transport, momentum transport, energy transport, etc.

The diffusion equation and the convection-diffusion equation are both parabolic PDEs.

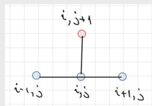
However, the presence of the first-order convection term has an influence on the numerical solution procedure.

42

42

The Forward-Time Centered-Space Method

$f_t = af_{xx} - bf_x$. Using i, j as the base point



f_t is approximated by the first-order forward-difference

$$f_t = \frac{f_{i,j+1} - f_{i,j}}{\Delta t}$$

f_x is approximated by the second-order centered-diff.

$$f_x = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x}$$

f_{xx} is approximated by the second-order centered- difference

$$f_{xx} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

The resulting FDE with $c = b\Delta t/\Delta x$ (convection number), $d = a\Delta t/\Delta x^2$

$$f_{i,j+1} = f_{i,j} - \frac{c}{2}\left(f_{i+1,j} - f_{i-1,j}\right) + d\left(f_{i+1,j} - 2f_{i,j} + f_{i-1,j}\right).$$

43

43

The Forward-Time Centered-Space Method

The forward-time centered-space method applied to the diffusion-convection equation is

- explicit
- two-level
- single-step
- $O(\Delta t) + O(\Delta x^2)$
- conditionally stable - criteria $c^2 \le 2d \le 1$ ($c = b\Delta t/\Delta x$, $d = a\Delta t/\Delta x^2$)
- convergent

Like most explicit methods applied to convection-diffusion equation, it is somewhat is inefficient, because the time step varies as the square of the spatial grid size $d = a\Delta t/\Delta x^2$

Additional criterion: for accuracy $R = \frac{b\Delta x}{a} \le 2$ (the cell Peclet or Reynolds number)

44

44

Other forward-time methods

Upwind method: first order $O(\Delta t) + O(\Delta x)$ (fast but not very good)

The Leonard method: $O(\Delta t) + O(\Delta x^2)$ some improvement to FTCS

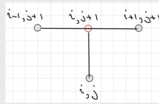
The DuFort-Frankel method: $O(\Delta t) + O(\Delta x^2) + O(\Delta t^2/\Delta x^2)$ explicit, three-level, single step and conditionally stable for $c \le 1$ (for any d). However, a starting method is required for the first step.

The MacCormack method: $O(\Delta t^2) + O(\Delta x^2)$ is explicit, two-level, two-step, conditionally stable. Excellent method for solving convection-diffusion problem. The method is applied even when b and a coefficients in $f_t = af_{xx} - bf_x$ are variable coefficients. The method is efficient for solving the non-linear convection-diffusion problem.

45

45

The Backward-Time Centered-Space Method



$f_t = af_{xx} - bf_x$. Using $i, j+1$ as the base point

f_t is approximated by the first-order forward-difference

$$f_t = \frac{f_{i,j+1} - f_{i,j}}{\Delta t}$$

f_x is approximated by the second-order centered-diff.

$$f_x = \frac{f_{i+1,j+1} - f_{i-1,j+1}}{2\Delta x}$$

f_{xx} is approximated by the second-order centered- difference

$$f_{xx} = \frac{f_{i+1,j+1} - 2f_{i,j+1} + f_{i-1,j+1}}{\Delta x^2}$$

The resulting FDE with $c = b\Delta t/\Delta x$ (convection number), $d = a\Delta t/\Delta x^2$

$$-\left(\frac{c}{2} + d\right) f_{i-1,j+1} + (1+2d) f_{i,j+1} + \left(\frac{c}{2} - d\right) f_{i+1,j+1} = f_{i,j}$$

46

46

The Backward-Time Centered-Space Method

The forward-time centered-space method applied to the diffusion-convection equation is

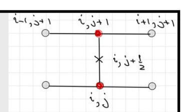
- implicit
- two-level
- single-step
- $O(\Delta t) + O(\Delta x^2)$
- unconditionally stable
- convergent

However, BTCS method becomes considerably more complicated when allied to non-linear PDEs, systems of PDEs and multidimensional problems.

47

47

The Crank-Nicolson method



The foundation of the method is the same as for the diffusion equation.

$$-\left(\frac{c}{2} + d\right) f_{i-1,j+1} + 2(1+d) f_{i,j+1} + \left(\frac{c}{2} - d\right) f_{i+1,j+1}$$
$$= \left(\frac{c}{2} + d\right) f_{i-1,j} + 2(1-d) f_{i,j} - \left(\frac{c}{2} - d\right) f_{i+1,j}$$

where $c = b\Delta t/\Delta x$ (convection number), and $d = a\Delta t/\Delta x^2$ (diffusion number). The method is implicit (for one-dimensional problems can be used with Thomas algorithm), two-level, single-step, $O(\Delta t^2) + O(\Delta x^2)$, unconditionally stable and convergent.

48

48

Other methods

The Keller box method: $O(\Delta t^2) + O(\Delta x^2)$, allows the physical grid spacing to be non-uniform.

The hopscotch method is a combination of FTCS and BTCS methods. It is generally used for two- and three-dimensional problems. The method is conditionally stable $c \leq 1$, and $O(\Delta t) + O(\Delta x^2)$

49

49

SUMMARY for the Parabolic PDEs

Explicit (FTCS) methods

- are conditionally stable and require a relatively small step size in marching direction to satisfy stability criteria.
- nonlinear problems and multidimensional problems can be solved directly by explicit methods

Implicit (BTCS) methods

- are unconditionally stable. The marching step size is restricted by accuracy requirements, not stability requirements
- for transient problems, the marching step size cannot be very much larger than for explicit methods

50

50