Numerical Methods for Parabolic Equations-I

Grétar Tryggvason
Spring 2010

Computational Fluid Dynamics I
http://users.wpi.edu/~gretar/me612.html

Solution Methods for Parabolic Equations

One-Dimensional Problems
- Explicit, implicit, Crank-Nicolson
- Accuracy, stability
- Various schemes

Multi-Dimensional Problems
- Alternating Direction Implicit (ADI)
- Approximate Factorization of Crank-Nicolson

Splitting
Stability in terms of fluxed

One-Dimensional Problems
- Explicit, implicit, Crank-Nicolson
- Accuracy, stability
- Various schemes

Multi-Dimensional Problems
- Alternating Direction Implicit (ADI)
- Approximate Factorization of Crank-Nicolson

Consider the diffusion equation
\[ \frac{\partial f}{\partial t} = \alpha \frac{\partial^2 f}{\partial x^2}, \quad t > 0, \quad a < x < b \]
which is a parabolic equation requiring

\[ f(x,0) = f_0(x) \quad \text{Initial Condition} \]
\[ f(a,t) = \phi_a(t); \quad f(b,t) = \phi_b(t) \quad \text{Boundary Condition (Dirichlet)} \]

or
\[ \frac{\partial f}{\partial x}(a,t) = \phi'_a(t); \quad \frac{\partial f}{\partial x}(b,t) = \phi'_b(t) \quad \text{Boundary Condition (Neumann)} \]

Parabolic equations can be viewed as the limit of a hyperbolic equation with two characteristics as the signal speed goes to infinity.

Increasing signal speed

Approximate the derivatives:
\[ \frac{\partial f}{\partial x} \left( \frac{f_{j+1}^{n+1} - f_j^n}{\Delta} \right) = \frac{\partial^2 f}{\partial x^2} \left( \frac{f_{j+1}^{n+1} - 2f_j^n + f_{j-1}^n}{\Delta^2} \right) \]
Explicit: FTCS

\[
f_{j+1} - f_j = \alpha \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}
\]

\[
f_{j+1} = f_j + \frac{\alpha \Delta t}{h^2} \left(f_{j+1} - 2f_j + f_{j-1}\right)
\]

Stability: von Neumann Analysis

\[
\frac{\epsilon^{n+1}}{\epsilon^n} = 1 - 4 \cdot \frac{\alpha \Delta t}{h^2} \sin^2 \left(\frac{\beta}{2}\right) \left[ G = 1 - 4r \sin^2 \left(\frac{\beta}{2}\right) \right]
\]

\[-1 < 1 - 4 \cdot \frac{\alpha \Delta t}{h^2} < 1
\]

\[0 \leq \frac{\alpha \Delta t}{h^2} < \frac{1}{2}\]

Fourier Condition

Implicit Method: Backward Euler

\[
f_{j+1} - f_j = \alpha \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}
\]

\[-rf_{j+1} + (2r + 1)f_j - rf_{j-1} = f_j \quad (r = \alpha \Delta t / h^2)
\]

Tri-diagonal matrix system

\[
\frac{\partial f}{\partial t} - \alpha \frac{\partial^2 f}{\partial x^2} = \frac{\alpha h^2}{12} (1 - 6r) f_{xx} + O(\Delta t^2, h^2, \Delta_x, h^4) f_{xxx}
\]

where \( r = \frac{\alpha \Delta t}{h^2} \)

- Accuracy \( O(\Delta t, h^2) \)
- If \( r = 1/6 \), then \( O(\Delta t^2, h^2) \)
- No odd derivatives; dissipative

Boundary effect is not felt at \( P \) for many time steps
This may result in unphysical solution behavior

Unconditionally stable
Stability Property

\[ \frac{1}{2} \leq \theta \leq 1 \quad \text{unconditionally stable} \]

\[ 0 \leq \theta < \frac{1}{2} \quad \text{stable only if} \]

\[ 0 \leq r \leq \frac{1}{2 - 4\theta} \]

Modified Equation

\[ \frac{\partial f}{\partial t} - \alpha \frac{\partial^2 f}{\partial x^2} = \frac{\alpha h^2}{12} f_{n+1} - \frac{\alpha h^2}{12} f_n + \frac{1}{360} \alpha^2 h^2 f_{n+1} - f_n \]

- Second-order accuracy \( O(\Delta x^2, h^3) \)

Amplification Factor (von Neumann analysis)

\[ G = \frac{1 - r (1 - \cos \beta)}{1 + r (1 - \cos \beta)} \]

Unconditionally stable

Combined Method A - 1

Generalization

\[ f_j^{n+1} - f_j^n = \alpha \left( f_j^{n+1} - 2f_j^n + f_j^{n-1} + (1 - \theta) f_j^{n+1} - 2f_j^n + f_j^{n+1} \right) \]

\[ \theta = \begin{cases} 0 & \text{Explicit (FTCS)} \\ 1 & \text{Implicit} \\ 1/2 & \text{Crank-Nicolson} \end{cases} \]

\[ n+1 \quad \times \theta \]

\[ n \]

\[ j-1 \quad j \quad j+1 \]

Modified Equation

\[ f_j^{n+1} - \alpha f_j^n = \left( \frac{\theta - 1}{2} \right) \frac{\partial^2 f}{\partial x^2} + \left( \frac{\theta - 1}{3} \right) 2 \Delta x^2 + \frac{1}{6} \left( \frac{\theta - 1}{2} \right) \left( \frac{1}{12} \right) \Delta x^2 + \frac{1}{360} \alpha \Delta x^2 \]

Special Cases

\[ \theta = \frac{1}{2} - \frac{1}{12r} \Rightarrow O(\Delta x^2, h^3) \]

\[ \theta = \frac{1}{2} - \frac{1}{12r} \quad r = 1/\sqrt{20} \Rightarrow O(\Delta x^2, h^3) \]

Combined Method A - 2

Generalized Three-Time-Level Implicit Scheme:

\[ (\theta + \theta) f_j^{n+1} - \theta f_j^n - f_j^{n+1} = \alpha \left( f_j^{n+1} - 2f_j^n + f_j^{n+1} \right) \]

\[ \theta = \begin{cases} 0 & \text{Explicit} \\ 1/2 & \text{Three-level fully implicit} \end{cases} \]

\[ n+1 \quad \times (\theta + \theta) \]

\[ n \]

\[ j-1 \quad j \quad j+1 \]

Combined Method B - 1
The Richardson method can be made stable by adding a term to the modified equation:

\[ f_{n+1} = (\frac{1}{2} - \frac{1}{2}) f_n + \frac{\Delta t}{12} f_{n+1} + O(\Delta t^2) + \ldots \]

Special Cases:
- \( \theta = \frac{1}{2} \) \( \Rightarrow \) Unstable
- \( \theta = \frac{1}{2} + \frac{1}{12r} \) \( \Rightarrow \) Conditionally stable

DuFort-Frankel - 1

The Richardson method can be made stable by splitting \( f_j^n \) by time average \( \left( f_j^{n+1} + f_j^{n-1} \right) / 2 \)

\[ f_j^{n+1} = f_j^{n-1} + 2r \left( f_j^n - f_j^{n-1} + f_j^{n+1} \right) \]

Diagrams:
- Richardson's Method: A Case of Failure
- DuFort-Frankel - 1
- DuFort-Frankel - 2

And others!
Consider a 2-D heat equation
\[ \frac{\partial f}{\partial t} = \alpha \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) \]

Two-dimensional grid

\[
\begin{align*}
\frac{f_{i,j}^{n+1} - f_{i,j}^{n}}{\Delta t} &= \alpha \left( f_{i+1,j}^{n} - 2f_{i,j}^{n} + f_{i-1,j}^{n} + f_{i,j+1}^{n} - 2f_{i,j}^{n} + f_{i,j-1}^{n} \right) \\
\text{If} \quad \Delta x = \Delta y &= h \\
\frac{f_{i,j}^{n+1} - f_{i,j}^{n}}{\Delta t} &= \frac{\alpha}{h^2} \left( f_{i+1,j}^{n} + f_{i-1,j}^{n} + f_{i,j+1}^{n} + f_{i,j-1}^{n} - 4f_{i,j}^{n} \right)
\end{align*}
\]
Isolate the new $f_i^*$ and solve by iteration

$$f_i^{n+1} = \frac{1}{1+4A}(A(f_i^{n+1} + f_{i+1}^{n+1} + f_{i-1}^{n+1} + f_{i,j}^{n+1}) + f_i^*)$$

The implicit method is unconditionally stable, but it is necessary to solve a system of linear equations at each time step. Often, the time step must be taken to be small due to accuracy requirements and an explicit method is competitive.

**Von Neumann Analysis**

$$e^{\Delta t} = e^{\Delta t}(e^{\Delta t} + e^{\Delta t} + e^{\Delta t} + e^{\Delta t} - 4)$$

$$e^{\Delta t} = 1 + \frac{\alpha \Delta t}{h^2}(2 \cos kh + 2 \cos mh - 4)$$

$$e^{\Delta t} = 1 - \frac{\alpha \Delta t}{h^2}\left(\frac{\sin kh}{2} + \frac{\sin mh}{2}\right)$$

Worst case

$$-1 \leq 1 - \frac{8\alpha \Delta t}{h^2} \leq 1 \quad \alpha \Delta t \leq \frac{1}{4}$$

**Stability limits depend on the dimension of the problems**

\[
\frac{\Delta t}{h^2} = \begin{cases} \frac{1}{2} & \text{One-dimensional flow} \\ \frac{1}{4} & \text{Two-dimensional flow} \\ \frac{1}{6} & \text{Three-dimensional flow} \end{cases}
\]

Different numerical algorithms usually have different stability limits.

Recall forward in time method

$$f_i^{n+1} = f_i^n + \left( \frac{\Delta t}{h^2} \right) f_i^{n+1} + f_{i+1}^{n+1} + f_{i-1}^{n+1} - 4 f_i^n$$

Evaluate the spatial derivatives at the new time \((n+1)\), instead of at \(n\)

\[
f_i^{n+1} = f_i^n + \left( \frac{\Delta t}{h^2} \right) f_i^n + f_{i+1}^{n+1} + f_{i-1}^{n+1} - 4 f_i^n
\]

This gives a set of linear equations for the new temperatures:

\[
(1 + 4A)f_i^{n+1} - A(f_i^{n+1} + f_{i+1}^{n+1} + f_{i-1}^{n+1} + f_{i,j}^{n+1}) = f_i^n
\]

Known source term

Second order accuracy in time can be obtained by using the Crank-Nicolson method.
Crank-Nicolson Method for 2-D Heat Equation

\[
\frac{f^{n+1} - f^n}{\Delta t} = \alpha \left( \frac{\partial^2 f^{n+1}}{\partial x^2} + \frac{\partial^2 f^{n+1}}{\partial y^2} \right) + \frac{\partial^2 f^n}{\partial x^2} + \frac{\partial^2 f^n}{\partial y^2} \]

If \( \Delta x = \Delta y = h \)

\[
f_{i,j}^{n+1} = f_{i,j}^{n} + \frac{\alpha \Delta t}{2h^2} \left( f_{i+1,j}^{n+1/2} + f_{i-1,j}^{n+1/2} + f_{i,j+1}^{n+1/2} + f_{i,j-1}^{n+1/2} - 4f_{i,j}^{n+1/2} \right) + \frac{\partial^2 f^n}{\partial x^2} + \frac{\partial^2 f^n}{\partial y^2} \]

The matrix equation is expensive to solve.

Expensive to solve matrix equations.

Can larger time-step be achieved without having solve the matrix equation resulting from the two-dimensional system?

The break through came with the Alternation-Direction-Implicit (ADI) method (Peaceman & Rachford-late 1950’s)

ADI consists of first treating one row implicitly with backward Euler and then reversing roles and treating the other by backwards Euler.

In matrix form, for each row

\[
\begin{bmatrix}
    f_{i,j}^{n+1/2} \\
    f_{i+1,j}^{n+1/2} \\
    \vdots \\
    f_{i+N,j}^{n+1/2}
\end{bmatrix}
= \begin{bmatrix}
    f_{i,j}^n \\
    f_{i+1,j}^n \\
    \vdots \\
    f_{i+N,j}^n
\end{bmatrix} + \frac{\alpha \Delta t}{h^2} \begin{bmatrix}
    -2 & 1 & \cdots & 0 \\
    1 & -2 & \cdots & \vdots \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & \cdots & 1 & -2
\end{bmatrix}
\begin{bmatrix}
    f_{i,j}^{n+1/2} \\
    f_{i+1,j}^{n+1/2} \\
    \vdots \\
    f_{i+N,j}^{n+1/2}
\end{bmatrix}
\]

This equation is easily solved by forward elimination and back-substitution.

Instead of solving one set of linear equations for the two-dimensional system, solve 1D equations for each grid line.

The directions can be alternated to prevent any bias.

\[\text{Computational Fluid Dynamics I}\]

Combining the two becomes equivalent to:

\[f^{n+1} = f^n + \alpha \Delta t \left[ \frac{\partial^2 f^{n+1}}{\partial x^2} + \frac{\partial^2 f^{n+1}}{\partial y^2} \right] + \frac{\partial^2 f^n}{\partial x^2} + \frac{\partial^2 f^n}{\partial y^2} \]

Step 1:  
\[f^{n+1/2} - f^n = \frac{\alpha \Delta t}{2h^2} \left( f_{i+1,j}^{n+1/2} - 2f_{i,j}^{n+1/2} + f_{i-1,j}^{n+1/2} + f_{i,j+1}^{n+1/2} - f_{i,j}^{n+1/2} \right) \]

Step 2:  
\[f^{n+1/2} - f^{n-1/2} = \frac{\alpha \Delta t}{2h^2} \left( f_{i+1,j}^{n+1/2} - 2f_{i,j}^{n+1/2} + f_{i,j+1}^{n+1/2} + f_{i,j}^{n-1/2} - 2f_{i,j}^{n+1/2} + f_{i,j}^{n+1/2} \right) \]

Combining the two becomes equivalent to:

\[f^{n+1} - f^n = \frac{\alpha \Delta t}{2} \left[ \frac{\partial^2 f^{n+1}}{\partial x^2} + \frac{\partial^2 f^{n+1}}{\partial y^2} \right] \]

Midpoint, Trapisodial

\[\text{Computational Fluid Dynamics I}\]

Fractional Step:  
\[\Delta x = \Delta y = h \]

\[f^{n+1} = f^n + \alpha \Delta t \left[ \frac{\partial^2 f^{n+1}}{\partial x^2} + \frac{\partial^2 f^{n+1}}{\partial y^2} \right] + \frac{\partial^2 f^n}{\partial x^2} + \frac{\partial^2 f^n}{\partial y^2} \]

\[\text{Computational Fluid Dynamics I}\]

\[\text{Computational Fluid Dynamics I}\]
ADI Method is \( O(\Delta t^2, h^2) \) accurate

Stability Analysis: \( e^{\epsilon} = e^{\epsilon_{even}} e^{\epsilon_{odd}} \)

\[
e^{\epsilon_{even}} = e^{\frac{\alpha \Delta t}{h} \sin \frac{kh}{2}} \left[ e^{\epsilon_{odd}} (e^{\Delta t - 2 + e^{\Delta t}}) + e^{\epsilon (e^{\Delta t} - 2 + e^{\Delta t})} \right]
\]

\[
e^{\epsilon_{odd}} = \frac{1 - 2 \frac{\alpha \Delta t}{h} \sin \frac{kh}{2}}{1 + 2 \frac{\alpha \Delta t}{h} \sin \frac{kh}{2}}
\]

Similarly,

\[
e^{\epsilon_{even}} = \frac{1 - 2 \frac{\alpha \Delta t}{h} \sin \frac{mh}{2}}{1 + 2 \frac{\alpha \Delta t}{h} \sin \frac{mh}{2}}
\]

Combining

\[
e^{\epsilon_{even}} \leq \left( \frac{1 - 2 \frac{\alpha \Delta t}{h} \sin \frac{kh}{2}}{1 + 2 \frac{\alpha \Delta t}{h} \sin \frac{kh}{2}} \right) \left( \frac{1 - 2 \frac{\alpha \Delta t}{h} \sin \frac{mh}{2}}{1 + 2 \frac{\alpha \Delta t}{h} \sin \frac{mh}{2}} \right) < 1
\]

Unconditionally stable!

The 3-D version does not have the same desirable stability properties. However, it is possible to generate similar methods for 3D problems

Implicit methods for parabolic equations

- Allow much larger time step (but must be balanced against accuracy!)
- Preserve the parabolic nature of the equation

But, require the solution of a linear set of equations and are therefore much more expensive than explicit methods

ADI provides a way to convert multidimensional problems into a series of 1D problems

Outline

Solution Methods for Parabolic Equations

- One-Dimensional Problems
  - Explicit, implicit, Crank-Nicolson
  - Accuracy, stability
  - Various schemes

- Multi-Dimensional Problems
  - Alternating Direction Implicit (ADI)
  - Approximate Factorization of Crank-Nicolson

Splitting

Numerical Methods for Parabolic Equations-III

Grétar Tryggvason
Spring 2010
Define $\delta_n$ by:
$$\delta_n = \left[ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 + 2 \delta_n \\ 1 \\ 1 \\ 1 + 2 \delta_n \\ 1 \\ 1 + 2 \delta_n \end{array} \right]$$

The ADI method can be written as
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

Eliminating $f^{n+1/2}$
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1/2} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

The Crank-Nicolson for heat equation becomes
$$f^{n+1} - f^n = \frac{\alpha}{2 \Delta t} f_0$$

which can be rewritten as
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

Eliminating $f^{n+1/2}$
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1/2} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

Define $\delta_n$ by:
$$\delta_n = \left[ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 + 2 \delta_n \\ 1 \\ 1 \\ 1 + 2 \delta_n \\ 1 \\ 1 + 2 \delta_n \end{array} \right]$$

The Crank-Nicolson for heat equation becomes
$$f^{n+1} - f^n = \frac{\alpha}{2 \Delta t} f_0$$

which can be rewritten as
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

Eliminating $f^{n+1/2}$
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1/2} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

Up to a factor:
$$\left( 1 - \frac{\alpha \Delta t}{2 \delta_n} \right) f^{n+1/2} = \left( 1 + \frac{\alpha \Delta t}{2 \delta_n} \right) f^n$$

ADI is an approximate factorization of the Crank-Nicolson method
Why splitting?

1. Stability limits of 1-D case apply.
2. Different $\Delta t$ can be used in $x$ and $y$ directions.

Implicit time marching by fast elliptic solvers

Stability from a flux point of view

Can be solved by elliptic solvers

Initial conditions

All other temperatures are unchanged

$$T_j^{n+1} = T_j^n + \frac{\Delta \alpha}{h^2} \left( T_{j+1}^n - 2 T_j^n + T_{j-1}^n \right)$$

$$T_j^{n+1} = T_j^n + \frac{\Delta \alpha}{h^2} \left( T_{j+1}^n + 2 T_j^n + T_{j-1}^n \right)$$

$$T_j^{n+1} = T_j^n + \frac{\Delta \alpha}{h^2} \left( 0 - 0 + 1 \right) = \frac{\Delta \alpha}{h^2}$$
One-dimensional unsteady diffusion by the FTCS scheme

```matlab
% one-dimensional unsteady diffusion by the FTCS scheme

n=40; nstep=300; length=2.0; h=length/(n-1); diff=0.05;
time=0; T=zeros(n,1); T(1)=1.0; dt=0.65*h^2/diff;
for m=1:nstep
    hold off; plot(T,'linewidt',2); axis([1 n -1.0, 1.0]);
    set(gca,'FontSize',24); set(gca,'LineWidth',2); pause;
    To=T;
    for i=2:n-1,
        T(i)=To(i)+diff*(dt/h^2)*(To(i+1)-2*To(i)+To(i-1));
        time=time+dt;
    end
end;
```

Multi-Dimensional Problems

- Alternating Direction Implicit (ADI)
- Approximate Factorization of Crank-Nicolson

Splitting

Stability in terms of fluxed