Computational Fluid Dynamics I
Elementary Numerical Analysis: Finite Difference Approximations-V
Grétar Tryggvason
Spring 2010

Consistency
Does the error always go to zero?

Consider the 1-D advection-diffusion equation
\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} = D \frac{\partial^2 f}{\partial x^2}
\]
and its finite-difference approximation
\[
\frac{f_{j+1}^n - f_j^n}{\Delta t} + U \frac{f_{j+1}^n - f_{j-1}^n}{2h} = D \frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{h^2}
\]
The discrepancy between the two equations can be found by deriving the modified equation.

Substituting
\[
\frac{f_{j+1}^n - f_j^n}{\Delta t} = \frac{\partial f(t)}{\partial t} + \frac{\partial^2 f(t)}{2 \partial t^2} + \ldots
\]
\[
\frac{f_{j+1}^n - f_{j-1}^n}{2h} = \frac{\partial f(x)}{\partial x} + \frac{\partial^3 f(x)}{6 \partial x^3} + \ldots
\]
\[
\frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{h^2} = \frac{\partial^2 f(x)}{\partial x^2} + \frac{\partial^4 f(x)}{12 \partial x^4} + \ldots
\]
into the finite difference equation
\[
\frac{f_{j+1}^n - f_j^n}{\Delta t} + U \frac{f_{j+1}^n - f_{j-1}^n}{2h} = D \frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{h^2}
\]
In this case, the error goes to zero as $h \to 0$ and $\Delta t \to 0$, so the approximation is said to be **consistent**.

Results in:

Original Equation

\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} - D \frac{\partial^2 f}{\partial x^2} = -\frac{\partial^2 f(t) \Delta t}{2} - U \frac{\partial^2 f(x) h^2}{6} + D \frac{\partial^2 f(x) h^2}{12} + \ldots
\]

Error terms

Shorthand:

\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} - D \frac{\partial^2 f}{\partial x^2} = O(\Delta t, h^2)
\]

Although most finite difference approximations are consistent, innocent-looking modifications can sometimes lead to approximations that are not!

The Frankel-Dufort is an example of a non-consistent scheme.

You will examine it in the homework.

Now modify it slightly:

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

Replace by:

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

This gives:

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

Which is easily solved for $f_j^n$ at the new time step in the HW, you will examine the error!

The **modified equation** is obtained by substituting the expression for the finite difference approximations, including the error terms, into the finite difference equation. For a **consistent** finite difference approximation the error terms go to zero as $h \to 0$ and $\Delta t \to 0$.

The modified equation can often be used to infer the nature of the error of the finite difference scheme. More about that later.

**Two-dimensional equation**
We will use the model equation:

\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} + V \frac{\partial f}{\partial y} = D \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)
\]

to demonstrate how to solve a partial equation (initial value problem) numerically.

The extension to two-dimensions is relatively straightforward, once the one-dimensional problem is fully understood.

For a two-dimensional flow discretize the variables on a two-dimensional grid

\[(x, y)\]

\[
\begin{align*}
  f_{i,j+1} &= f(x,y+h) \\
  f_{i,j} &= f(x,y) \\
  f_{i+1,j} &= f(x+h,y)
\end{align*}
\]

Solve for \(f_{i,j}^{n+1}\)

\[
f_{i,j}^{n+1} = f_{i,j}^n + \Delta t \left\{ -U \left( \frac{f_{i+1,j}^n - f_{i-1,j}^n}{2h} \right) - V \left( \frac{f_{i,j+1}^n - f_{i,j-1}^n}{2h} \right) + \right. \\
&\quad \left. \frac{D}{h^2} \left[ f_{i+1,j}^n + f_{i-1,j}^n + f_{i,j+1}^n + f_{i,j-1}^n - 4f_{i,j}^n \right] \right\}
\]

or

\[
f_{i,j}^{n+1} = f_{i,j}^n - \frac{\Delta U}{2h} (f_{i+1,j}^n - f_{i-1,j}^n) - \frac{\Delta V}{2h} (f_{i,j+1}^n - f_{i,j-1}^n) + \frac{\Delta t D}{h^2} (f_{i+1,j}^n + f_{i,j+1}^n + f_{i,j-1}^n + f_{i-1,j}^n - 4f_{i,j}^n)
\]

Accuracy: \(O(\Delta t, \Delta x^2)\)

A stability analysis gives:

\[
\frac{\Delta t D}{\Delta x^2} \leq \frac{1}{2} \quad \text{and} \quad \frac{\Delta U |V \Delta x^2 | \Delta t}{D} \leq 4
\]

Example

\[
\frac{\partial f}{\partial t} + U \frac{\partial f}{\partial x} + V \frac{\partial f}{\partial y} = D \left( \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)
\]

Uniform flow through the domain

\[U=1 \quad V=0\]
\[ f_{i,j}^{n+1} = f_{i,j}^n - \frac{\Delta t}{2h} \left( f_{i+1,j}^n - f_{i-1,j}^n \right) + \frac{\Delta D}{h^2} \left( f_{i+1,j}^n + f_{i,j+1}^n + f_{i,j-1}^n - 4f_{i,j}^n \right) + \Delta t U \frac{f_{i+1,j}^n + f_{i,j+1}^n + f_{i,j-1}^n - 4f_{i,j}^n}{h^2} \]

\[ \frac{\partial f}{\partial n} \text{ or } f \text{ given on the boundary} \]

\[ \frac{\partial f}{\partial y} = 0 \]

At the \( i=1 \) boundary, for example,

\[ \frac{\partial f}{\partial y} \approx f_{i,2}^n - f_{i,1}^n = 0 \]

we find that:

\[ f_{i,2}^n = f_{i,1}^n \]
Consider the Poisson Equation:
\[ \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = S \]

This equation has a solution if \( f \) or \( \frac{\partial f}{\partial n} \) is specified on the boundary.

Use standard finite differences to discretize:
\[ \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2} = S_{i,j} \]

Solve for \( f_{i,j} \):
\[ f_{i,j} = \frac{1}{4} \left( f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - h^2 S_{i,j} \right) \]

The iteration must be carried out until the solution is sufficiently accurate. To measure the error, define the residual:
\[ R_{i,j} = \frac{f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j}}{h^2} - S_{i,j} \]

At steady-state the residual should be zero. The pointwise residual or the average absolute residual can be used, depending on the problem. Often, simpler criteria, such as the change from one iteration to the next is used.

Although the Jacobi iteration is a very robust iteration technique, it converges VERY slowly.

We therefore seek a way to ACCELERATE the convergence to steady-state, making use of the fact that it is only the steady-state that is of interest.

Here we introduce the Gauss-Seidler method and the Successive Over-Relaxation (SOR) method.
The Jacobi iteration can be improved somewhat by using new values as soon as they become available.

\[ f_{i,j}^{a+1} = \frac{1}{4} \left( f_{i+1,j}^a + f_{i-1,j}^a + f_{i,j+1}^a + f_{i,j-1}^a - h^2 S_{i,j} \right) \]

From a programming point of view, Gauss-Seidler iteration is even simpler than Jacobi iteration since only one vector with \( f \) values is needed.

The Gauss-Seidler iteration can be accelerated even further by various acceleration techniques. The simplest one is the Successive Over-Relaxation (SOR) iteration

\[ f_{i,j}^{a+1} = \frac{\beta}{4} \left( f_{i+1,j}^a + f_{i-1,j}^a + f_{i,j+1}^a + f_{i,j-1}^a + h^2 S_{i,j} \right) + (1-\beta) f_{i,j}^a \]

The SOR iteration is very simple to program, just as the Gauss-Seidler iteration. The user must select the coefficient. It must be bounded by \( 1 < \beta < 2 \). \( \beta = 1.5 \) is usually a good starting value.

Example

\[ \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0 \]

% two-dimensional steady-state problem by SOR
n=40; m=40; iterations=5000; length=2.0; h=length/(n-1);
T=zeros(n,m); bb=1.7;
T(10:n-10,1)=1.0;
for l=1:iterations,
    for i=2:n-1, for j=2:m-1
        T(i,j)=bb*0.25*(T(i+1,j)+T(i,j+1)+T(i-1,j)+T(i,j-1))+(1.0-bb)*T(i,j);
    end,end
    % find residual
    res=0;
    for i=2:n-1, for j=2:m-1
        res=res+abs(T(i+1,j)+T(i,j+1)+T(i-1,j)+T(i,j-1)-4*T(i,j))/h^2;
    end,end
    if rem((n-2)*(m-2)) == 0
        fprintf('Success: iteration and residual: \( (n-2) \times (m-2) = 0 \)\n
The program is easily modified for the Jacobi and the Gauss-Seidler iteration:

Average absolute error: 0.001
Number of iterations
Jacobi: 1989
Gauss-Seidler: 986
SOR (\( \beta = 1.5 \)): 320
SOR (\( \beta = 1.7 \)): 162
SOR (\( \beta = 1.9 \)): 91
SOR (\( \beta = 1.95 \)): 202
The converged solution: 
\[
\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0
\]

A Finite Difference Code for the Navier-Stokes Equations in Vorticity/Streamfunction Form

Grétar Tryggvason
Spring 2010

Objectives

- The Driven Cavity Problem
- The Navier-Stokes Equations in Vorticity/Streamfunction form
- Boundary Conditions
- The Grid
- Finite Difference Approximation of the Vorticity/Streamfunction equations
- Finite Difference Approximation of the Boundary Conditions
- Iterative Solution of the Elliptic Equation
- The Code
- Results
- Convergence Under Grid Refinement

Outline

- Developing an understanding of the steps involved in solving the Navier-Stokes equations using a numerical method
- Write a simple code to solve the "driven cavity" problem using the Navier-Stokes equations in vorticity form

The vorticity/streamfunction equations:

\[
\begin{align*}
\frac{\partial}{\partial t} \left( \frac{\partial \omega}{\partial x} \right) + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} &= \frac{1}{Re} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \\
\frac{\partial}{\partial t} \left( \frac{\partial v}{\partial x} \right) + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{\partial p}{\partial y} + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \\
\frac{\partial}{\partial t} \left( \frac{\partial u}{\partial y} \right) + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\
\omega &= \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}
\end{align*}
\]
Computational Fluid Dynamics I

The vorticity/streamfunction equations:

Solve the incompressibility conditions

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

by introducing the stream function

$$u = \frac{\partial \psi}{\partial y}; \quad v = -\frac{\partial \psi}{\partial x}$$

Substituting:

$$\frac{\partial}{\partial x} \left( \frac{\partial \psi}{\partial y} \right) - \frac{\partial}{\partial y} \left( \frac{\partial \psi}{\partial x} \right) = 0$$

Substituting

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}$$

into the definition of the vorticity

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

yields

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega$$

The Navier-Stokes equations in vorticity-stream function form are:

Advection/diffusion equation

$$\frac{\partial \omega}{\partial t} = -\frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y} + 1 \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) + \text{Re} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right)$$

Elliptic equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega$$

Boundary Conditions for the Streamfunction

At the right and the left boundary:

$$u = 0 \Rightarrow \frac{\partial \psi}{\partial y} = 0$$

$$\Rightarrow \psi = \text{Constant}$$

At the top and the bottom boundary:

$$v = 0 \Rightarrow -\frac{\partial \psi}{\partial x} = 0$$

$$\Rightarrow \psi = \text{Constant}$$

Since the boundaries meet, the constant must be the same on all boundaries:

$$\psi = \text{Constant}$$
The normal velocity is zero since the streamfunction is a constant on the wall, but the zero tangential velocity must be enforced:

At the right and left boundary: \(|v| = 0 \Rightarrow -\frac{\partial \psi}{\partial x} = 0\)
At the top boundary: \((u = 0 \Rightarrow \frac{\partial \omega}{\partial y} = 0)\)
At the bottom boundary: \((u = U_{wall} \Rightarrow \frac{\partial \omega}{\partial y} = U_{wall})\)

Similarly, at the top and the bottom boundary:
\[
\frac{\partial \psi}{\partial x} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad \Rightarrow \quad \omega_{wall} = -\frac{\partial^2 \psi}{\partial y^2}
\]

To compute an approximate solution numerically, we start by laying down a discrete grid:

\[\psi_{i,j} \text{ and } \omega_{i,j}\]
stored at each grid point

\[\text{Grid boundaries coincide with domain boundaries}\]

Then we replace the equations at each grid point by a finite difference approximation:

\[
\frac{\partial \omega}{\partial x}_{i,j}^{n} = \frac{\partial \psi}{\partial y}_{i,j}^{n} + \frac{\partial \psi}{\partial x}_{i,j}^{n} + \frac{1}{Re} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right)_{i,j}^{n}
\]

\[
\frac{\partial^2 \psi}{\partial x^2}_{i,j}^{n} + \frac{\partial^2 \psi}{\partial y^2}_{i,j}^{n} = -\omega_{i,j}^{n}
\]
The elliptic equation is:

\[
\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega
\]

\[
\psi_{i+1,j}^{n} + \psi_{i-1,j}^{n} + \psi_{i,j+1}^{n} + \psi_{i,j-1}^{n} - 4\psi_{i,j}^{n} = -\omega_{i,j}^{n}
\]
Consider the bottom wall ($j=1$):

Need to find $\omega_{\text{wall}} = \omega_{i,j=1}$

Given:

$\psi = \text{Constant}$

$\frac{\partial \psi}{\partial y} = U_{\text{wall}}$; $\omega_{\text{wall}} = -\frac{\partial \psi}{\partial y}$

Expand the streamfunction

$\psi_{i,j=2} = \psi_{i,j=1} + \frac{\partial \psi_{j=1}}{\partial y} h + \frac{\partial^2 \psi_{j=1}}{2 \partial y^2} h^2 + O(h^3$)

Using:

$\omega_{\text{wall}} = \frac{\partial^2 \psi_{j=1}}{2 \partial y^2}$; $U_{\text{wall}} = \frac{\partial \psi_{j=1}}{\partial y}$

this becomes:

$\psi_{i,j=2} = \psi_{i,j=1} + U_{\text{wall}} h - \omega_{\text{wall}} \frac{h^2}{2} + O(h^3)$

Solving for the wall vorticity:

$\omega_{\text{wall}} = \left( \psi_{i,j=1} - \psi_{i,j=2} \right) \frac{2}{h^2} + U_{\text{wall}} \frac{2}{h} + O(h)$

The elliptic equation:

$\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4\psi_{i,j} = -\frac{h^2}{h^2}$

Rewrite as

$\psi_{i,j}^{n+1} = 0.25 \left( \psi_{i+1,j}^{n} + \psi_{i-1,j}^{n} + \psi_{i,j+1}^{n} + \psi_{i,j-1}^{n} + h^2 \omega_{i,j}^{n} \right)$

Solve by SOR

$\psi_{i,j}^{n+1} = \beta 0.25 \left( \psi_{i+1,j}^{n} + \psi_{i-1,j}^{n} + \psi_{i,j+1}^{n} + \psi_{i,j-1}^{n} + h^2 \omega_{i,j}^{n} \right)$

+ $\left( 1 - \beta \right) \psi_{i,j}^{n}$

Limitations on the time step

$$\frac{u \Delta t}{h^2} \leq \frac{1}{4} \quad \frac{(u + 1) \Delta t}{v} \leq 2$$
for i=1:MaxIterations
    for i=2:nx-1; for j=2:ny-1
        \( s(i,j) = \text{SOR for the stream function} \)
    end; end
end
for i=2:nx-1; for j=2:ny-1
    \( \text{rhs}(i,j) = \text{Advection+diffusion} \)
end; end
Solve for the stream function
Find vorticity on boundary
Find RHS of vorticity equation
Initial vorticity given
\( t = t + \Delta t \)
Update vorticity in interior
\( v(i,j) = \text{...} \)
\( v(i,j) = v(i,j) + \Delta t \times \text{rhs}(i,j) \)

### Results:

17 by 17
\( \Delta t = 0.01 \)
\( D = 0.1 \)

Vorticity at \( t = 1.2 \)
Streamfunction at \( t = 1.2 \)

17 by 17
\( \Delta t = 0.01 \)
\( D = 0.1 \)

Velocity
Streamfunction at \( t = 1.2 \)
Vorticity at \( t = 1.2 \)