Numerical Methods for Partial Differential Equations

CAAM 452
Spring 2005
Lecture 2
Instructor: Tim Warburton
Note on textbook for finite difference methods

• Due to the difficulty some students have experienced in obtaining Gustafsson-Kreiss-Oliger I will try to find appropriate and equivalent sections in the online finite-difference book by Trefethen:

http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/pdetext.html
Recall Last Lecture

- We considered the model advection PDE:

$$\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = 0$$

- defined on the periodic interval $[0, 2\pi)$

- We recalled that any $2\pi$ periodic, $C^1$ function could be represented as a uniformly convergent Fourier series, so we considered the evolution of the PDE with a single Fourier mode as initial condition. This converted the above PDE into a simpler ODE for the time-dependent coefficient (i.e. amplitude) of a Fourier mode:

$$\frac{d\hat{u}}{dt} - i\omega c\hat{u} = 0$$
Recall: the Advection Equation

- We will start with a specific Fourier mode as the initial condition:

1) Find $2\pi$-periodic $u(x,t)$ such that $\forall x \in [0,2\pi), t \in [0,T]$

$$\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = 0$$

given

$$u(x,0) = f(x) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{f}(\omega) \quad \forall x \in [0,2\pi)$$

where $f$ is a smooth $2\pi$-periodic function of one frequency $\omega$

- We try to find a solution of the same type:

$$u(x,t) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{u}(\omega,t)$$
Substituting in this type of solution the PDE:

\[
\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = 0
\]

Becomes an ODE:

\[
\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = \left( \frac{\partial}{\partial t} - c \frac{\partial}{\partial x} \right) \left( \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{u}(\omega, t) \right) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \left( \frac{d\hat{u}}{dt} - i\omega c \hat{u} \right)
\]

\[\Rightarrow \frac{d\hat{u}}{dt} - i\omega c \hat{u} = 0\]

With initial condition

\[\hat{u}(\omega, 0) = \hat{f}(\omega)\]
• We have Fourier transformed the PDE into an ODE.
• We can solve the ODE:

\[
\begin{align*}
\frac{d\hat{u}}{dt} - i\omega c \hat{u} &= 0, \\
\hat{u}(\omega, 0) &= \hat{f}(\omega)
\end{align*}
\]

\[\Rightarrow \hat{u}(\omega, t) = e^{i\omega ct} \hat{u}(\omega, 0) = e^{i\omega ct} \hat{f}(\omega)\]

• And it follows that the PDE solution is:

\[
\text{ansatz: } u(x, t) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{u}(\omega, t)
\]

\[
\text{solution: } \hat{u}(\omega, t) = e^{i\omega ct} \hat{f}(\omega)
\]

\[\Rightarrow u(x, t) = \frac{1}{\sqrt{2\pi}} e^{i\omega(x+ct)} \hat{f}(\omega) = f(x + ct)\]

initial condition: \[f(x) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{f}(\omega)\]
Note on Fourier Modes

• Note that since the function should be 2pi periodic we are able to deduce: \( \omega \in \mathbb{Z} \)

• We can also use the superposition principle for the more general case when the initial condition contains multiple Fourier modes:

\[
f(x) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\omega=\infty} e^{i\omega x} \hat{f}(\omega)
\]

\[
\Rightarrow u(x,t) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\omega=\infty} e^{i\omega(x+ct)} \hat{f}(\omega) = f(x+t)
\]
• Let’s back up a minute – the crucial part was when we reduced the PDE to an ODE:

$$\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = 0 \quad \rightarrow \quad \frac{d\hat{u}}{dt} - i\omega c\hat{u} = 0$$

• The advantage is: we know how to solve ODE’s both analytically and numerically (more about this later on).
Add Diffusion Back In

- So we have a good handle on the advection equation, let’s reintroduce the diffusion term:

\[
\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = d \frac{\partial^2 u}{\partial x^2}
\]

- Again, let’s assume 2-pi periodicity and assume the same ansatz:

\[
u = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{u}(\omega, t)
\]

- This time:

\[
\frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = d \frac{\partial^2 u}{\partial x^2}
\]

\[
\frac{d\hat{u}}{dt} - i\omega c\hat{u} = -d\omega^2 \hat{u}
\]

\[
\frac{d\hat{u}}{dt} = (ic\omega - d\omega^2) \hat{u}
\]
• Again, we can solve this trivial ODE:

\[
\frac{d\hat{u}}{dt} = (i\omega c - d \omega^2) \hat{u}
\]

\[
\hat{u} = \hat{u}(\omega, 0) e^{(i\omega - d \omega^2)t}
\]

\[
u = \hat{u}(\omega, 0) e^{i\omega(ct + x)} e^{-d \omega^2 t}
\]
The solution tells a story:

\[ u = \hat{u}(\omega, 0) e^{i\omega (ct+x)} e^{-d\omega^2 t} \]

The original profile travels in the direction of decreasing \( x \) (first term)

As the profile travels it decreases in amplitude (second exponential term)
What Did Diffusion Do??

- **Advection:**
  \[
  \frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = 0 \quad \rightarrow \quad \frac{d\hat{u}}{dt} - i\omega c\hat{u} = 0
  \]

- **Diffusion:**
  \[
  \frac{\partial u}{\partial t} - c \frac{\partial u}{\partial x} = d \frac{\partial^2 u}{\partial x^2} \quad \rightarrow \quad \frac{d\hat{u}}{dt} = (i\omega c - d\omega^2)\hat{u}
  \]

- Adding the diffusion term shifted the multiplier on the right hand side of the Fourier transformed PDE (i.e. the ODE) into the left half plane.

- We summarize the role of the multiplier…
Categorizing a Linear ODE

Here we plot the dependence of the solution to the top left ODE on μ’s position in the complex plane.

\[
\frac{d\hat{u}}{dt} = \mu \hat{u} \implies \hat{u} = \hat{u}(0) e^{\mu t}
\]
Categorizing a Linear ODE

Here we plot the behavior of the solution for 5 different choices of \( \mu \)

CAAM 452 Spring 2005
Summary

\[ \frac{d\hat{u}}{dt} = \mu \hat{u} \quad \Rightarrow \quad \hat{u} = \hat{u}(0) e^{\mu t} = \hat{u}(0) e^{\text{Re}(\mu)t} (\cos(\text{Im}(\mu)t) + i \sin(\text{Im}(\mu)t)) \]

- When the real part of \( \mu \) is negative the solution decays exponentially fast in time (rate determined by the magnitude of the real part of \( \mu \)).

- When the real part of \( \mu \) is positive the solution grows exponentially fast in time (rate determined by the magnitude of the real part of \( \mu \)).

- If the imaginary part of \( \mu \) is non-zero the solution oscillates in time.

- The larger the imaginary part of \( \mu \) is, the faster the solution oscillates in time.
Solving the Scalar ODE Numerically

• We know the solution to the scalar ODE

\[
\frac{d\hat{u}}{dt} = \mu \hat{u}
\]

• However, it is also reasonable to ask if we can solve it approximately.

• We have now simplified as far as possible.

• Once we can solve this model problem numerically, we will apply this technique using the method of lines to approximate the solution of the PDE.
Stage 1: Discretizing the Time Axis

\[
\frac{du}{dt} = f(u)
\]

- It is natural to divide the time interval \([0,T]\) into shorted subintervals, with width often referred to as: 
  \(dt, \Delta t\) or \(k\)

- We start with the initial value of the solution \(u(0)\) (and possibly \(u(-dt), u(-2dt), ..\)) and build a recurrence relation which approximates \(u(dt)\) in terms of \(u(0)\) and early values of \(u\).
Example

• Using the following approximation to the time derivative:

\[
\frac{du}{dt} \approx \frac{u(t + dt) - u(t)}{dt}
\]

• We write down an approximation to the ODE:

\[
\frac{du}{dt} = f(u) \quad \Rightarrow \quad \frac{u(dt) - u(0)}{dt} \approx f(u(0))
\]
Example cont

• Rearranging:

\[
\frac{u(dt) - u(0)}{dt} \approx f(u(0))
\]

\[u(dt) \approx u(0) + dtf(u(0))\]

• We introduce notation for the approximate solution after the n’th time step: \( u_n \)
• Our intention is to compute \( u_n \approx u(ndt) \)
• We convert the above equation into a scheme to compute an approximate solution:

\[
\begin{align*}
    u_0 &= u(0) \\
    u_1 &= u_0 + dtf(u_0)
\end{align*}
\]
Example cont

• We can repeat the following step from $t=dt$ to $t=2dt$ and so on:

\[
\begin{align*}
    u_0 &= u(0) \\
    u_1 &= u_0 + dt f(u_0) \\
    u_2 &= u_1 + dt f(u_1) \\
    & \vdots \\
    u_{n+1} &= u_n + dt f(u_n)
\end{align*}
\]

• This is commonly known as the:
  – Euler-forward time-stepping method
  – or Euler’s method
Euler-Forward Time Stepping

• It is natural to ask the following questions about this time-stepping method:

\[ u_0 = u(0) \]

\[ u_{n+1} = u_n + dt f(u_n) \]

• Does the answer get better if \( dt \) is reduced? (i.e. we take more time-steps between \( t=0 \) and \( t=T \))

• Does the numerical solution behave in the same way as the exact solution for general \( f \)?
  (for the case of \( f(u) = mu^2u \) does the numerical solution decay and/or oscillate as the exact solution)

• How close to the exact solution is the numerical solution?

• As we decrease \( dt \) does the end iterate converge to the solution at \( t=T \)
Experiments

• Before we try this analytically, we can code it up and see what happens.

• This is some matlab code for Euler forward applied to:

\[
\begin{align*}
    u_0 &= u(0) \\
    u_{n+1} &= u_n + dt \mu u_n
\end{align*}
\]

% purpose: solve du/dt = mu*u
%    u(0) = u0
% for t in [0,T] with intervals of dt

% input:
%  T = final time
%  dt = time step
%  u0 = initial value of solution
%  mu = multiplier for ODE rhs
% output:
%  u = vector of values (u_0, u_1, ..., u_nsteps)
%  dt = actual time step used

function [u,t] = EulerForwardODE(T, dt, u0, mu)

    nsteps = ceil(T/dt);
    dt = T/nsteps;
    u = zeros(nsteps+1,1);
    t = zeros(nsteps+1,1);
    mu = -1-2*sqrt(-1);

    u(1) = u0;
    t(1) = 0;
    for n=1:nsteps
        f = mu*u(n);
        u(n+1) = u(n) + dt*f;
        t(n+1) = t(n) + dt;
    end

http://www.caam.rice.edu/~caam452/CodeSnippets/EulerForwardODE.m
What dt can we use?

- With dt=0.1, T=3, uo=1, mu=-1-2*i

The error is quite small.
Larger $dt=0.5$

- With $dt=0.5$, $T=3$, $u_0=1$, $\mu=-1-2i$

The error is visible but the trend is not wildly wrong.
Even Larger $dt=1$

- With $dt=1$, $T=3$, $u_0=1$, $\mu=-1-2*i$

The solution is not remotely correct – but is at least bounded.
• With $T=30$, $dt=2$, $u_0=1$, $\mu=-1-i$

Boom – the approximate solution grows exponentially fast, while the true solution decays!
Observations

• When dt is small enough we are able to nicely approximate the solution with this simple scheme.

• As dt grew the solution became less accurate.

• When dt=1 we saw that the approximate solution did not resemble the true solution, but was at least bounded.

• When dt=2 we saw the approximate solution grew exponentially fast in time.
• Our observations indicate two qualities of time stepping we should be interested in:
  
  – stability: i.e. is the solution bounded in a similar way to the actual solution?

  – accuracy: can we choose dt small enough for the error to be below some threshold?
Geometric Interpretation:

• We can interpret Euler-forward as a shooting method.
• We suppose that $u_n$ is the actual solution, compute the actual slope $f(u_n)$ and shift the approximate solution by $dt f(u_n)$.

Note:
- the blue line is not the exact solution, but rather the same ODE started from the last approximate value of $u$ computed.
- in this case we badly estimated the behavior of even the approximately started solution in the interval $dt$.
→ two kinds of error can accumulate!
Interpolation Interpretation:

- We start with the ODE: \( \frac{du}{dt} = f(u) \)

- Integrate both sides from over a \( dt \) interval:
  \[
  \int_{t_n}^{t_{n+1}} \frac{du}{dt} \, dt = \int_{t_n}^{t_{n+1}} f(u) \, dt
  \]

- Use the fundamental theorem of calculus:
  \[
  \int_{t_n}^{t_{n+1}} \frac{du}{dt} \, dt = [u]_{t_n}^{t_{n+1}} = u_{n+1} - u_n
  \]

- Finally, replace \( f \) with a constant which interpolates \( f \) at the beginning of the interval...
\[ \int_{t_n}^{t_{n+1}} \frac{du}{dt} \, dt \bigg|_{t_n}^{t_{n+1}} = u(t_{n+1}) - u(t_n) \]

\[ \int_{t_n}^{t_{n+1}} f(u(t)) \, dt \equiv \int_{t_n}^{t_{n+1}} f(u(t_n)) \, dt = f(u(t_n)) \int_{t_n}^{t_{n+1}} dt = f(u(t_n)) \, dt \]

\[ \Rightarrow u(t_{n+1}) - u(t_n) \equiv f(u(t_n)) \, dt \]

Again, we have recovered Euler’s method.
Let’s choose $f(u) = mu^*u$

$u_0 = u(0)$

$u_{n+1} = u_n + dt\mu u_n = (1 + dt\mu)u_n$

We can solve this immediately:

$u_0 = u(0)$

$u_{n+1} = (1 + dt\mu)u_n = (1 + dt\mu)^{n+1}u(0)$

Next suppose $\text{Re}(\mu) \leq 0$ then we expect the actual solution to be bounded in time by $u(0)$. For this to be true of the approximate solution we require: $|1 + dt\mu| \leq 1$
Stability Condition for Euler Forward

- Since \( \mu \) is fixed we are left with a condition which must be met by \( dt \)

\[
|1 + dt \mu| \leq 1
\]

- which is true if and only if:

\[
(1 + \text{Re}(\mu dt))^2 + (\text{Im}(\mu dt))^2 \leq 1
\]

- The region of the complex plane which satisfies this condition is the interior of the unit circle centered at \(-1+0*i\)
i.e. $\mu dt$ will not blow up if it is located inside the unit disk:

Notice: the exact solutions corresponding to $\text{Re}(\mu t)<0$ all decay. However, only the numerical solutions corresponding to the interior of the yellow circle decay.
Disaster for Advection!!!

• Now we should be worried. The stability region:

![Diagram showing complex plane with point 
\(-1+0i\) on the imaginary axis.]

• only includes one point on the imaginary axis (the origin) **but** our advection equation for the periodic interval has \( \mu \) which are purely imaginary!!!.

\[
\frac{d\hat{u}}{dt} - i\omega c \hat{u} = 0 \implies \mu = i\omega c
\]

• Conclusion – the Euler-Forward scheme applied to the Fourier transform of the advection equation will generate exponentially growing solutions.
Nearly a Disaster for Advection-Diffusion!!!

• This time the stability region must contain:

\[ \mu dt = (i\omega c - \omega^2 d) dt \]

• Because the eigenvalues are shifted into the left half part of the complex plane, we will be able to choose \( dt \) small enough to force the \( \mu dt \) into the stability region.

• We can estimate how small \( dt \) has to be in this case
cont (estimate of dt for advection-diffusion)

• For stability we require:

\[ |\mu dt + 1| \leq 1 \]

\[ \iff |(i\omega c - \omega^2 d) dt + 1| \leq 1 \]

\[ \iff (1 - dt\omega^2 d)^2 + (\omega c dt)^2 \leq 1 \]

\[ \iff -2dt\omega^2 d + (dt\omega^2 d)^2 + (\omega c dt)^2 \leq 0 \]

\[ \iff dt = 0 \text{ or } \omega = 0 \text{ or } dt(\omega^2 d^2 + c^2) \leq 2d \]
The only interesting condition for stability is:

\[ dt \leq \frac{2d}{\omega^2 d^2 + c^2} \]

What does this mean?, volunteer?

i.e. how do \( \omega, d, c \) influence the maximum stable \( dt \)?

Hint: consider

i) \( d \) small, \( c \) large

ii) \( d \) large, \( c \) small
Multistep Scheme (AB2)

- Given the failure of Euler-Forward for our goal equation, we will consider using the solution from 2 previous time steps.

- i.e. we consider \[ u_{n+1} = u_n + dt\left( af(u_n) + bf(u_{n-1}) \right) \]

- Recall the integral based interpretation of Euler-Forward:

\[
u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} f(u(t)) \, dt \approx f(u(t_n)) \, dt\]
cont

- We interpolate through \( f(u_{n-1}) \) and \( f(u_n) \) then integrate between \( t_n \) and \( t_{n+1} \)

\[
\text{Linear interpolant of the f values}
\]

\[
\text{Approximate integral}
\]
This time we choose to integrate under the interpolant of $f$ which agrees with $f$ at $t_n$ and $t_{n-1}$.

The unique linear interpolant in this case is:

$$I_1f = \left(\frac{t-t_{n-1}}{t_n-t_{n-1}}\right)f(u_n) + \left(\frac{t-t_n}{t_{n-1}-t_n}\right)f(u_{n-1})$$

We need to compute the following integral of the interpolant:

$$\int_{t_n}^{t_n} I_1f \, dt$$
\[ \int_{t_n}^{t_{n+1}} I_{1,f} dt = \int_{t_n}^{t_{n+1}} \left( \frac{t-t_n}{t_n-t_{n-1}} \right) f(u_n) + \left( \frac{t-t_n}{t_{n-1}-t_n} \right) f(u_{n-1}) dt \]

\[ = \frac{f(u_n)}{dt} \int_{t_n}^{t_{n+1}} (t-t_{n-1}) dt - \frac{f(u_{n-1})}{dt} \int_{t_n}^{t_{n+1}} (t-t_n) dt \]

\[ = \frac{f(u_n)}{dt} \left[ \frac{t^2}{2} - t_{n-1}t \right]_{t_n}^{t_{n+1}} - \frac{f(u_{n-1})}{dt} \left[ \frac{t^2}{2} - t_nt \right]_{t_n}^{t_{n+1}} \]

\[ = dt \left( \frac{3}{2} f(u_n) - \frac{1}{2} f(u_{n-1}) \right) \]

Last step: homework exercise
The resulting AB2 scheme requires the solution at two levels to compute the update:

\[ u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} I f_1 dt \]

\[ = u_n + \left( \frac{3}{2} f(u_n) - \frac{1}{2} f(u_{n-1}) \right) \]
Next Time

- Stability region for AB2
- Generalization to AB3, AB4 (use more historical data to compute update)
- Accuracy/consistency of Euler-Forward, AB2, AB3, AB4
- Convergence.
- Runge-Kutta schemes (if time permits)

**READING:**

http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/1all.pdf

- p10-p55 (do not do exercises)