Math 405/607E, Section 101, Fall 2014

Numerical Methods for Differential Equations

Instructor: Brian Wetton, MATX 1107, wetton@math.ubc.ca

Web Page: www.math.ubc.ca/~wetton/

Audience: The course is intended for 3rd and 4th year students in Science or Engineering who wish to learn the basic numerical techniques they will require in business, industry, or graduate school. The course will also be useful to graduate students who have not taken basic numerical methods courses as part of their undergraduate training and who need to learn these skills in order to do their research.

Undergraduate Prerequisites: Math 405 has a prerequisite of one of Math 256, 257, or 316.

Graduate Credit: The graduate version (Math 607E) of this course has a prerequisite of some knowledge of differential equations (ordinary and partial). A project involving some more detailed numerical analysis or computation is required in addition to the undergraduate material.

Course Objectives: The primary objective of the course is to introduce the basic numerical techniques for solving ordinary and partial differential equations in a single course, which does not require any previous numerical courses as a prerequisite. The basic numerical methods (e.g. interpolation, numerical integration, numerical differentiation, numerical linear algebra and root finding) are introduced and then applied to the solution of ordinary and partial differential equations. This approach helps to contextualize the numerical methods and enables us to focus on applications of the methods to practical problems.

Text: No text. Written notes will be provided. Some suggestions of optional texts will be provided on the web page.

Material: Newton's method for nonlinear problems. Functional approximation by power series, piecewise polynomial and spectral techniques. Numerical integration and differentiation. Discretization techniques for differential equation boundary value problems: finite difference, finite element and spectral methods. Fast solution techniques: direct

sparse solvers and iterative methods. Time stepping techniques for initial value problems. Computational implementation is an important aspect of the course.

Marks (Math 405): 40% final, 10% midterm and 50% assignments

Marks (Math 607E): 30% final, 10% midterm, 40% assignments and 20% final project

Midterm Date: Thursday, October 16 in class.

Assignments: There will be five challenging assignments. Some computation will be required. MATLAB is a high level mathematical computation package that is suitable for these computations, but other packages or basic computer languages can be used.

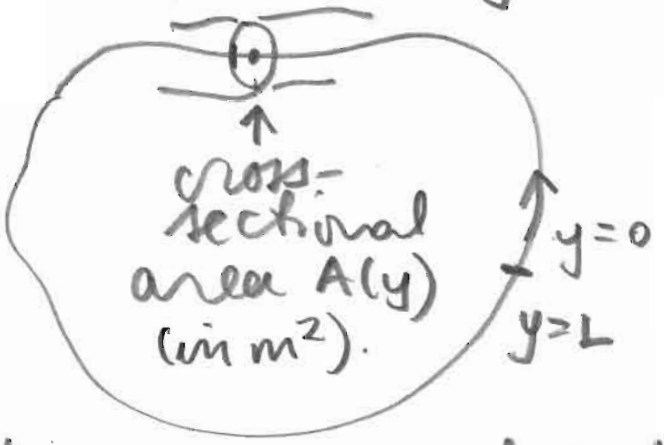
Project: Required for the graduate version (Math 607E) of the course. Topics will be finalized the week after the midterm in consultation with the instructor. The project could be a computation related to the student's thesis work.

Scaling & Nondimensionalization
Example.

Brian Wetton, wetton@math.ubc.ca

January 7, 2013.

Consider heat conduction in a metal rod of length L (in m) bent around with its two ends joined.



Assume that the aspect ratio of the rod is large enough that the temperature $V(y,t)$ (in °C) can be assumed to be constant in the cross-section.

Local heat balance gives the following

$$A\rho c\, V_t = (KAV_y)_y + f(y,t) - g(V,V_0,y) \qquad (1)$$

where:

ρ: density of the rod material kg/m³

c: heat capacity J/kg/°C

K: thermal conductivity J/s/m/°C

F: given applied heating per unit length
J/s/m.

g: heat loss per unit length to ambient
J/s/m. This has a functional form that must be fit to experiments. We allow a dependence on y since this could be affected by the cross-section shape.
clearly, $g(V_0, V_0, y) = 0$ for every y.

Note: Every termin (1) has the same units, J/s/m. Checking for unit consistency is a useful step in modelling.

Now let's make some assumptions: the external heating $f(y)$ does not depend on t and we consider the system after a long enough time that transients have died away, so $V(y)$. Also assume that A is constant and g does not depend on y. The final assumption at this stage is that a linear Taylor approx of $g(V, V_0) \approx M(V - V_0)$
is sufficient (M has units of $\frac{J}{sm^\circ C}$)

M is determined experimentally or through asymptotics of the surface heat transfer to the ambient medium.

Now (1) becomes

$$-KA\frac{d^2V}{dy^2} + M(V-V_0) = f(y). \qquad (2)$$

Scale spatial variables

$$y = Lx, \qquad x \in [0,1], \text{ dimensionless.}$$

and shift V, $w = V - V_0$, so (2) becomes.

$$-\frac{KA}{L^2}\frac{d^2W}{dx^2} + MW = f(xL). \qquad (3)$$

Let F (in J/s/m) be a representative size of f, so that

$$\frac{f(xL)}{F} := \hat{f}(x)$$

is dimensionless and has unit size, $O(1)$.

Scale w,

$$w = Vu, \quad u \text{ dimensionless}$$

with V in °C to be determined. (3) becomes

$$-\frac{KAV}{L^2}u'' + MVu = F\hat{f}(x) \Rightarrow$$

$$-u'' + \frac{ML^2}{KA} u = \frac{FL^2}{KAV} \hat{f}(x)$$

So if we choose the temperature scale to be

$$V = \frac{FL^2}{KA}$$

dimensionless parameter.

and let $a = ML^2/KA$, we arrive at our scaled problem

$$-u'' + au = \hat{f}(x).$$

in which all quantities are dimensionless.

Some Notes on the Fourier Transform

Brian Wetton

November 25, 2012

1 Fourier Series on a finite interval

Consider a function $f(x)$ defined and periodic on the interval $[0, 2\pi]$. We can expand the function in fourier series as follows:

$$f(x) = \frac{1}{2}a_0 + \sum_1^\infty [a_n \cos nx + b_n \sin nx]. \tag{1}$$

The coefficients a_n and b_n that make this "work" are

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos nx dx \tag{2}$$

$$b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin nx dx. \tag{3}$$

That these are the coefficients that we must take is seen from the fact that the functions $\{1, \sin x, \cos x, \sin 2x, \cos 2x \ldots\}$ are orthogonal in the following sense:

$$\int_0^{2\pi} \cos nx \cos mx dx = \begin{cases} 0 \text{ if } m \neq n \\ \pi \text{ if } m = n \neq 0 \\ 2\pi \text{ if } m = n = 0 \end{cases}$$

$$\int_0^{2\pi} \sin nx \sin mx dx = \begin{cases} 0 \text{ if } m \neq n \\ \pi \text{ if } m = n \end{cases}$$

$$\int_0^{2\pi} \cos nx \sin mx dx = 0 \text{ for all } m, n$$

These results can be easily verified. We can now multiply eq. (1) by $\sin mx$ or $\cos mx$ and integrate from 0 to 2π and use the results above to see that the coefficients must be given by eq (2, 3).

Note 1 *I'm hoping this is all old news for you.*

1.1 Validity

So far we know that if the expansion (1) is going to work, the coefficients must be given by (2, 3). Under what conditions do we actually get convergence of the right hand side of (1) and then when do we have equality in (1)? I will quote two results without proof:

1.1.1 Piecewise continuous functions

Suppose that $f(x)$ is a piecewise continuously differentiable function. This means that $f(x)$ is continuous and has a continuous derivative at all but a finite number of points, where it may suffer a jump (but remains bounded). For instance, the function

$$f(x) = \begin{cases} 1, & 0 \leq x < \pi \\ -1, & \pi \leq x < 2\pi \end{cases} \tag{4}$$

is a piecewise continuously differentiable function. Note that this function has jumps at π and at 0 (or 2π) since when we say differentiable we mean it in the sense of 2π periodicity.

We now state our theorem:

Theorem 1 *If f is a piecewise continuously differentiable function then the series on the right of eq. (1) converges at all x to*

$$\frac{1}{2}[f(x+0) + f(x-0)]$$

where $f(x+0)$ is the limit of $f(y)$ as y approaches x from the right and $f(x-0)$ is the limit of $f(y)$ as y approaches x from the left.

At all points where f is continuous (away from the jumps), $f(x+0) = f(x-0)$ and the series converges to $f(x)$.

The series associated with the function in eq. (4) converges to

$$\begin{cases} 1, & 0 < x < \pi \\ 0, & x = 0, \pi, 2\pi \\ -1, & \pi < x < 2\pi \end{cases}$$

1.1.2 Square integrable functions

Now consider f to be in the space of functions, $L_2[0, 2\pi]$. This space contains all functions defined on $[0, 2\pi]$ that are *measurable* (see [2] for example for this technical

definition but let me say any functions we will deal with are measurable) and such that

$$\int_0^{2\pi} |f(x)|^2 dx$$

is finite. Notice that this allows some unbounded functions as well as the piecewise smooth functions above.

Theorem 2 *If $f \in L_2$ then the fourier series of f converges to f in the following sense:*

$$\lim_{N\to\infty} \int_0^{2\pi} |f(x) - \frac{a_0}{2} - \sum_{n=1}^{N} [a_n \cos nx + b_n \sin nx]|^2 dx = 0.$$

This certainly does not guarantee pointwise convergence.

1.2 Complex form

For notational convenience (but this is very important as you'll see) it's much nicer to abandon the real form of the fourier series for the equivalent complex one. Remember the complex exponential

$$e^{(x+iy)} = e^x(\cos x + i \sin y),$$

one of the first complex analytic functions you learned about. This complex exponential obeys all of the normal rules for exponentials, *i.e.*

$$e^{z_1} \times e^{z_2} = e^{z_1+z_2}$$

for any complex numbers z_1 and z_2. Switching the roll of x above we get

$$e^{ix} = \cos x + i \sin x.$$

Now define the following complex fourier coefficients

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx} dx \tag{5}$$

for $n = 0, \pm 1, \pm 2, \ldots$. Clearly,

$$c_n = \begin{cases} \frac{a_0}{2}, & n = 0 \\ \frac{1}{2}(a_n - ib_n), & n > 0 \\ \frac{1}{2}(a_n + ib_n), & n < 0 \end{cases}$$

It is then possible to rewrite eq. (1) as

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}.$$

We think of c_n as being the fourier coefficients of $f(x)$ and so can naturally label them as $\hat{f}_n$ or $\mathcal{F}(f)_n$.

1.3 Normalization and alternate forms

There are several other ways to define the transform that you may see in other texts. The various forms have merely to do with the normalization and conjugation of the transform coefficients. Some other valid transform and inverse transform pairs are

$$\hat{f}_n = \int_0^{2\pi} f(x)e^{-inx}dx \qquad f(x) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} \hat{f}_n e^{inx}$$

$$\hat{f}_n = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{inx}dx \qquad f(x) = \sum_{n=-\infty}^{\infty} \hat{f}_n e^{-inx}$$

$$\hat{f}_n = \frac{1}{\sqrt{2\pi}}\int_0^{2\pi} f(x)e^{-inx}dx \qquad f(x) = \frac{1}{\sqrt{2\pi}}\sum_{n=-\infty}^{\infty} \hat{f}_n e^{inx}$$

This last form is particularly nice and is the one I use when I can. Its symmetric weighting leads to the nice property

$$\int_0^{2\pi} |f(x)|^2 dx = \sum_{n=-\infty}^{\infty} |\hat{f}_n|^2.$$

Note 2 *It is clear that fourier series can be applied to functions in any finite interval by scaling and translation.*

Question 1 *Consider functions defined on the interval $[a, b]$. Find a valid transform and inverse transform formula for this case.*

Question 2 *Find the fourier transform of the function in eq. (4).*

Question 3 *Find the fourier transform of the following function:*

$$f(x) = \begin{cases} x, & 0 \le x < \pi \\ 2\pi - x, & \pi \le x < 2\pi \end{cases}$$

2 Fourier Transform on the Real Line

When $f(x)$ is defined for all x the situation with the fourier transform and its inverse is a bit different. Here, the transform takes on a continuous number of values

$$\hat{f}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\alpha x} dx$$

for all real α and the inverse is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\alpha) e^{i\alpha x} d\alpha \tag{6}$$

As with fourier series, there are alternate forms with normalization and conjugation. Again, this is a convenient form since here

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{f}(x)|^2 dx.$$

Also, the same kind of validation arguments apply in this case (note that some kind of decay in f has to be assumed as $|x| \to \infty$ for the transform of a piecewise smooth function to even exist, though).

2.1 Some Properties of the Fourier Transform

2.1.1 Derivatives

If f has a derivative with a "valid" fourier transform, then the fourier transform of the derivative is given by

$$\mathcal{F}(f')(\alpha) = i\alpha \mathcal{F}(f)(\alpha).$$

This result can be found formally just by differentiating eq. (6). Here we have used $\mathcal{F}(f)$ instead of $\hat{f}$ to avoid confusion with the primes for derivatives. This is a marvelous result. In words, it says that the fourier transform reduces differentiation to multiplication. It is no wonder that fourier transform and series are used heavily in the study of differential equations.

Question 4 *Suppose $f(x)$ is a real-valued function on the whole real line with a well-defined fourier transform. Let $u(x)$ be the function that satisfies*

$$-u'' + u = f$$

at every point x (and u has suitable decay as $|x| \to \infty$). Find the relationship between $\hat{u}$ and $\hat{f}$.

2.1.2 Convolution

The convolution of two functions on the real line is given by

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x-s)g(s)ds$$

The fourier transform of a convolution is extremely simple:

$$\mathcal{F}(f * g)(\alpha) = \hat{f}(\alpha) \times \hat{g}(\alpha)$$

It reduces to pointwise multiplication of the transforms. The opposite is also true: the transform of a product is the convolution of their transforms.

Question 5 *In question # 4 the solution should be of the form*

$$\hat{u}(\alpha) = \hat{G}(\alpha)\hat{f}(\alpha).$$

*That is, the solution u is the convolution of f with a function G (known as the Greens function of the problem). Find an explicit form for $G(x)$ (*Hint: *G should have a discontinuous derivative at $x = 0$.*

3 Discrete Fourier Transform

We now consider the corresponding situation on a finite lattice ($\mathbf{R}^n$ or $\mathbf{C}^n$). For a N vector U with components U_j (index j from 0 to $N-1$ here) the DFT is a complex N vector $\hat{U}$ with components given by

$$\hat{U}_\alpha = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} U_j e^{-2\pi(ij\alpha)/N} \tag{7}$$

for $\alpha = 0, 1, \ldots N-1$. The inverse transform is given by

$$U_j = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \hat{U}_\alpha e^{2\pi(ij\alpha)/N} \tag{8}$$

The transform (7) is just a fancy way of describing the change of basis of the vector U to the ortho-normal coordinates $\{E_\alpha\}$ where $(E_\alpha)_j := e^{2\pi(ij\alpha)/N}/\sqrt{N}$.

As always, these formulas are arbitrary up to conjugation and scaling. With the scaling above we have

$$\|U\| = \|\hat{U}\|$$

where $\|\cdot\|$ is the usual Euclidean norm (which can be denoted $\|\cdot\|_2$ to distinguish it from other norms on $\mathbf{C}^n$),

$$\|U\| = \sqrt{\sum_{j=0}^{N-1} |U_j|^2}$$

This is just a consequence of the fact that the DFT is an orthonormal change of basis (such changes are Euclidean norm preserving).

Question 6 *Another norm for $\mathbf{C}^n$ is the maximum norm $\|\cdot\|_\infty$ given by*

$$\|U\|_\infty = \max_{j=0,1,\ldots N-1} |U_j|$$

Show that the DFT does not preserve maximum norm.

Question 7 *Find constants $c_1(N)$ and $c_2(N)$ such that*

$$c_1(N)\|U\|_2 \leq \|U\|_\infty \leq c_2(N)\|U\|_2$$

for all N-vectors U.

3.1 Some Linear Algebra

The operation in eq. (7) is equivalent to multiplication by a matrix which we will call $\mathcal{F}$. The operation in eq. (8) must then be described by multiplication by $\mathcal{F}^{-1}$. By inspection we see that $\mathcal{F}^{-1} = \mathcal{F}^*$ where $*$ denotes the Hermitian (or conjugate) transpose. This result follows from the fact that the DFT is just an orthonormal change of coordinates.

Note 3 *Multiplication by an $N \times N$ matrix is in general an operation that takes $O(N^2)$ computational work. However, the DFT in eq. (7) can be done faster than this and stably (in $O(N \log N)$ work) using the Fast Fourier Transform (FFT) method. This algorithm has made it possible to compute approximations using fourier transform ideas.*

3.2 Transform on an Infinite Lattice

We can come full circle back to section 1 by considering the transform of vectors on an infinite lattice, *i.e.* U_j with $j = 0, \pm 1, \pm 2, \ldots$ (with suitable decay as $|j| \to \infty$). In this case, the transform $\hat{u}(\alpha)$ takes on values for continuous values of α and $\hat{u}$ is 2π-periodic. We have

$$U_j = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \hat{u}(\alpha) e^{ij\alpha} d\alpha$$

with

$$\hat{u}(\alpha) = \frac{1}{\sqrt{2\pi}} \sum_{j=-\infty}^{\infty} U_j e^{-ij\alpha}.$$

This is just the complex Fourier Series from section 1 written backwards where integration is taken over the interval $[-\pi, \pi]$ instead of $[0, 2\pi]$ (equivalent since the integrand is 2π-periodic). This change allows correct interpretation of $\hat{u}(\alpha)$ values.

Note 4 *The DFT plays the same role in the analysis of finite difference equations as the FT plays in the analysis of differential equations. In this role, the use of the DFT is known as von Neumann analysis.*

Note 5 *I owe much of my understanding and many of the details in these notes to [1].*

References

[1] Marvin Shinbrot, "The Solution of Partial Differential Equations," unpublished lecture notes.

[2] Royden, "Real Analysis".

Math 405, Fall 2014, Day 1

Introduction

- go over outline
- show Fluent video

The fluids in the video are modelled (described approximately) by partial differential equations. **Modelling** ↑

The equations are for

$\rho(x,y,z,t)$ density

$\underline{u}(x,y,z,t)$ velocity $\quad \underline{u} = (u,v,w)$

$\varphi(x,y,z,t)$ volume fraction

($\varphi = 0$ blue liquid, $\varphi = 1$ red liquid)

Can't store a continuum function $\rho(x,y,z,t)$ on a computer, so it is discretized, that is, it is only stored at a finite set of points. The PDEs of the model are approximated using only the values at these points. ↑ **Discretization**

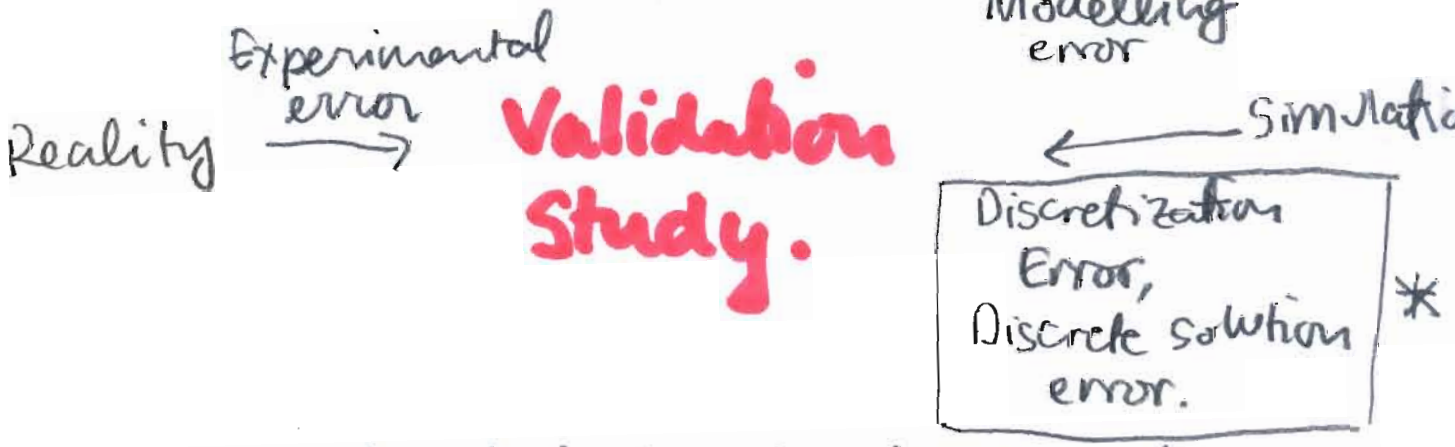
The discrete equations relate the quantities at one discrete time to the values at the next discrete time. This is a nonlinear system in which a large (millions +) number of unknowns are coupled together.

this nonlinear system is solved (approximately), often with an iterative method.

Discrete Solution

The results are then visualized and design decisions can be made. It is often faster and cheaper to use simulations to make design decisions rather than an experimental process.

However, you can't just imagine that the simulations are accurate enough to base design decisions on – they have to be validated by at least limited experiments.



It is important to be able to clearly identify a modelling error. To do this, we have to be able to be able to reduce discrete errors (*) to arbitrarily

low levels. For a fixed discretization, the discrete solver should get arbitrarily close to the solution of the discretization as the number of iterations increases. The discretization should become arbitrarily close to the PDE solution as the number of discrete points increases. Both these results are called convergence (of the iterative solver and the discretization respectively).

Convergence ↑

The object of this course is to introduce you to basic ideas of discretization, iterative solution of systems of equations, and convergence of methods to approximate DEs. There are some beautiful mathematical results, and you will see some of the ingredients of codes you might use in a future career (like Fluent). It is a start of the training to develop your own methods for new models.

We'll start with scalar, algebraic problems, then move to systems of algebraic equations, then increase the

complexity by going from discrete problems to ODE's, then PDE's. Before starting with ODEs, we'll warm-up with numerical interpolation, integration and differentiation

---

Brief discussion of Floating point approximation

IEEE standard for double precision FP representation of real numbers gives about 16 digits of accuracy in the range of about 10^{-308} to 10^{308}.

We can write

$$x \approx x(1+c\varepsilon) \qquad |c| \le 1, \quad \varepsilon = 10^{-16}$$

x: the real number we want

$x(1+c\varepsilon)$: FP approximation

x, error $|xc\varepsilon|$

The relative error $\dfrac{|xc\varepsilon|}{|x|} \le 10^{-16}$ ← **Important Definition**

There are two cases where FP accuracy may be an issue.

A When you subtract two numbers that are approximately equal, the relative error in the result can be large.

$(x+b) - x = b$ exact.

Consider $|b| \ll |x|$.

↑ much less than

$|c|, |d| \leq 1$.

FP

$$(x + b + xc\varepsilon) - (x + xd\varepsilon)$$

$$= b + x(c-d)\varepsilon.$$

relative error $\leq 2\varepsilon \underbrace{|x/b|}$

large amplification

Ex $x^2 - 2x + \gamma = 0$ $\quad |\gamma| \ll 1$.

roots $x = 1 \pm \sqrt{1-\gamma}$

$x_1 = 1 + \sqrt{1-\gamma}$ fine r.

$x_2 = 1 - \sqrt{1-\gamma}$ phenomena A, loss of accuracy.

Note, the formula for x_2 can be modified to avoid this problem,

$$x_2 = 1 - \sqrt{1-\gamma} \times \frac{1+\sqrt{1-\gamma}}{1+\sqrt{1-\gamma}}$$

$$= \frac{1-(1-\gamma)}{1+\sqrt{1-\gamma}} = \frac{\gamma}{1+\sqrt{1-\gamma}}$$ fine ✓.

↑ could have been deduced from γ/x_2.

B Ill conditioned linear systems.

$A\underline{x} = \underline{b}$ exact. ($n \times n$ system).

A $n \times n$, invertible, $\underline{x}$ & $\underline{b}$ n vectors in column.

Solved symbolically by $\underline{x} = A^{-1}\underline{b}$.

Solved in practice by Gaussian Elimination.

In MATLAB, $x = A \backslash b$. ← **Direct Solve** (no iteration).

$(\hat{A} + \varepsilon|B|)(\underline{x} + \varepsilon \underline{c}) = \underline{b} + \varepsilon \underline{d}$. FP approx.

Q: How large is $\underline{c}$ relative to $\underline{x}$?

First, how will we measure the size of a vector.

↑ Many choices of vector norms. Let's use the maximum norm for now.

$$\|\underline{x}\|_\infty = \max_{1 \le i \le n} |x_i|.$$

So $\|\underline{c}\|_\infty \le \|\underline{x}\|_\infty$ (relative error ε in each component).

$\|\underline{d}\|_\infty \le \|\underline{b}\|_\infty$

Every vector norm induces a matrix norm

$$\|\hat{A}\|_\infty := \max_{\underline{x} \neq \underline{0}} \frac{\|\hat{A}\underline{x}\|_\infty}{\|\underline{x}\|_\infty}$$

Important Definition

So $\|\hat{A}\underline{x}\|_\infty \le \|\hat{A}\|_\infty \|\underline{x}\|_\infty$

for every $\underline{x}$ and ↑ this is the smallest constant that makes this work.

In words, $\|\hat{A}\|_\infty$ is the largest amount that multiplying by $\hat{A}$ can increase the $\|\cdot\|_\infty$ of a vector.

So if $\mathbb{A}\underline{x} = \underline{b}$ (exact solution),

$$\|\underline{b}\|_\infty \leq \|\mathbb{A}\|_\infty \|\underline{x}\|_\infty$$

and so $\|\underline{d}\|_\infty \leq \|\mathbb{A}\|_\infty \|\underline{x}\|_\infty$.

To proceed, we need the following result, the proof of which will be left to an assignment question

to be shown

$$\|\mathbb{B}\|_\infty \leq \|\mathbb{A}\|_\infty$$ (component relative error leads to matrix max norm matrix relative error).

Now apply $\mathbb{A}^{-1}$ to the FP approximation at the top of page 7:

$$\varepsilon\underline{c} = (\underline{x} - \mathbb{A}^{-1}\underline{b}) + \varepsilon\|\mathbb{A}^{-1}\|\mathbb{B}\underline{x} + \varepsilon^2\|\mathbb{A}^{-1}\|\mathbb{B}\underline{c}$$

exact solution $+ \varepsilon\|\mathbb{A}^{-1}\underline{d}$ ↑ neglect ε^2, good approx unless $\mathbb{A}$ is extremely ill conditioned.

So now $\|\underline{c}\|_\infty \leq \|\mathbb{A}^{-1}\|_\infty \|\mathbb{A}\|_\infty \|\underline{x}\|_\infty$.

Recall that $\varepsilon\underline{c}$ is the error made in $\underline{x}$, so the relative error is

$$\frac{\varepsilon\|\underline{c}\|_\infty}{\|\underline{x}\|_\infty} \leq \varepsilon\|\mathbb{A}^{-1}\|_\infty \|\mathbb{A}\|_\infty$$

important Definition

condition number of $\mathbb{A}$.

Condition number $K(A)$ is a measure of the loss of accuracy when solving systems with coefficient matrix A.

Notes: (a) $K(A) \geq 1$. Matrices with K very large are called 'ill-conditioned.

(b) In the analysis above, we considered that the FP approximation at the top of page 7 was solved exactly. In practice, additional FP errors are made in the solution process. These can be reduced by pivoting strategies. The full analysis is complicated and the general result overestimates the resulting errors in most systems coming from applications.

(c) For ill-conditioned systems, a technique called iterative refinement can be used to increase solution accuracy.

↑ reference.

Math 405, Fall 2014, Day 1 Addendum

These notes should replace the arguments on page 728 of the original Day 1 notes. There is an error in the argument described there, and several students had problems following that first description. However, the definitions of matrix norm and condition numbers on those pages are correct.

Big picture FP accuracy (double precision) is more than enough for any application I can think of. FP approximation errors are typically much smaller than the other errors we will study in the class (discretization errors, iterative solver errors). We will neglect FP errors in the analysis of these other errors.

However, FP errors cannot always be neglected. We are considering two extreme cases (subtracting numbers of almost the same size and ill-conditioned systems) where they can be significant.

$$\mathbb{A}\underline{x} = \underline{b} \qquad (\mathbb{A}_{n\times n},\ \text{invertible}) \qquad (1)$$

Let us consider the effect of FP accuracy on the solution of this system in an idealized setting. Suppose $\underline{b}$ is approximated using FP and also the entries of $\mathbb{A}$. We idealize the effect by considering the resulting system solved by exact arithmetic.

$$(\mathbb{A} + \varepsilon \mathbb{E})\underline{y} = \underline{b} + \varepsilon \underline{d} \qquad (2).$$

We have

$$\|\underline{d}\|_\infty \le \|\underline{b}\|_\infty$$

relative accuracy $\varepsilon = 10^{-16}$

$$\|\mathbb{E}\|_\infty \le \|\mathbb{A}\|_\infty \quad \text{(assignment question)}.$$

Q: what is the relative accuracy of $\underline{y}$ to the exact $\underline{x}$?

Write $\underline{y} = \underline{x} + \varepsilon \underline{c}$ without loss of generality. Insert into (2) and apply $\mathbb{A}^{-1}$ to both sides

$$\underbrace{\mathbb{A}^{-1}(\mathbb{A} + \varepsilon \mathbb{E})}_{\mathbb{I}}(\underline{x} + \varepsilon \underline{c}) = \underbrace{\mathbb{A}^{-1}\underline{b}}_{\underline{x}} + \varepsilon \mathbb{A}^{-1}\underline{d}.$$

neglect, size ε^2

$$\cancel{\underline{x}} + \varepsilon \underline{c} + \varepsilon \mathbb{A}^{-1}\mathbb{E}\underline{x} + \cancel{\varepsilon^2 \mathbb{A}^{-1}\mathbb{E}\underline{c}} = \cancel{\underline{x}} + \varepsilon \mathbb{A}^{-1}\underline{d}$$

matrix norms, ↓ triangle inequality

$$\text{so } \underline{c} = -/\!A^{-1}/\!E\underline{x} + /\!A^{-1}\underline{d}.$$

$$\|\underline{c}\|_\infty \le \underbrace{\|/\!A^{-1}\|_\infty}_{\le \|/\!A\|_\infty} \|/\!E\|_\infty \|\underline{x}\|_\infty + \underbrace{\|/\!A^{-1}\|_\infty}_{\le \|b\|_\infty \le \|/\!A\|_\infty \|\underline{x}\|_\infty} \|\underline{d}\|_\infty$$

$$\text{so } \|\underline{c}\|_\infty \le 2 \underbrace{\|/\!A^{-1}\|_\infty \|/\!A\|_\infty}_{\text{condition number } K(/\!A)} \|\underline{x}\|_\infty$$

Since $\varepsilon\underline{c}$ is the error in $\underline{x}$, the relative error is

$$\frac{\varepsilon\|\underline{c}\|_\infty}{\|\underline{x}\|_\infty} \le 2\varepsilon K(/\!A).$$

If $\underline{x}$ were known and approximated to FP accuracy, the relative error would be ε. Thus, $2K(/\!A)$ is an amplification of the error due to the structure of the linear system.

Math 405, Fall 2014, Day 2

Consider nonlinear, algebraic problems – starting with scalar root finding. That is, finding values (roots) X_* for which

$$f(X_*) = 0$$

↑ f is a given differentiable (and so continuous) function. We will consider two iterative methods to approximate X_*. Both generate a sequence of estimates

$$X_0, X_1, \ldots X_n, \ldots$$

such that $\lim_{n\to\infty} X_n = X_*$. We say that $\{X_n\}$ converges to X_*.

important definition.

Recall our objective to compute quantities to arbitrarily high precision. ✓

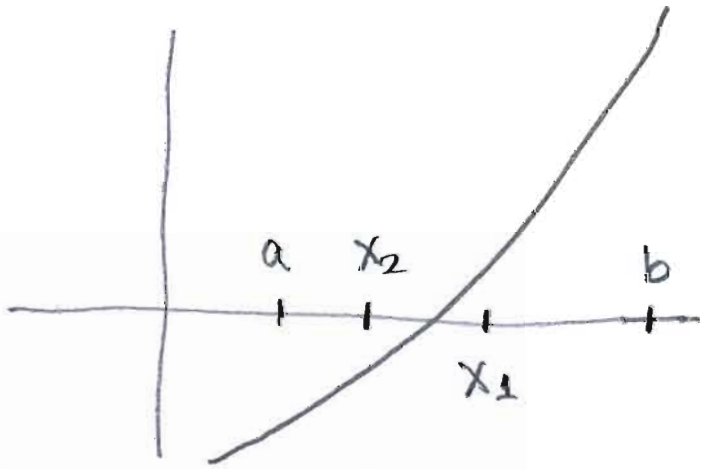
Consider $f(x)$ in the interval $x \in [a,b]$. If $f(a)$ and $f(b)$ have opposite signs $[f(a)\cdot f(b) < 0]$ then by the intermediate value theorem, f must have a root in the interval $[a,b]$. (It might have more than one).

Bisection method

Begin with a & b that bracket a root as described above.

Compute $x_1 = \frac{a+b}{2}$ (midpoint).

If $f(a) \cdot f(x_1) < 0$ then replace b with x_1, otherwise replace a with x_1. In this way, $[a,b]$ has been reduced in size but still brackets the root.



Then compute $x_2 = (a+b)/2$ and repeat. At each step, the size of the interval that a root must be in is reduced by a factor of 2.

Thus $\underbrace{|x_n - x_*|}_{\text{error } e_n} < \underbrace{\frac{b-a}{2^n}}_{\text{error bound } B_n}$.

This proves that the procedure converges, that is $\lim_{n\to\infty} x_n = x_*$.

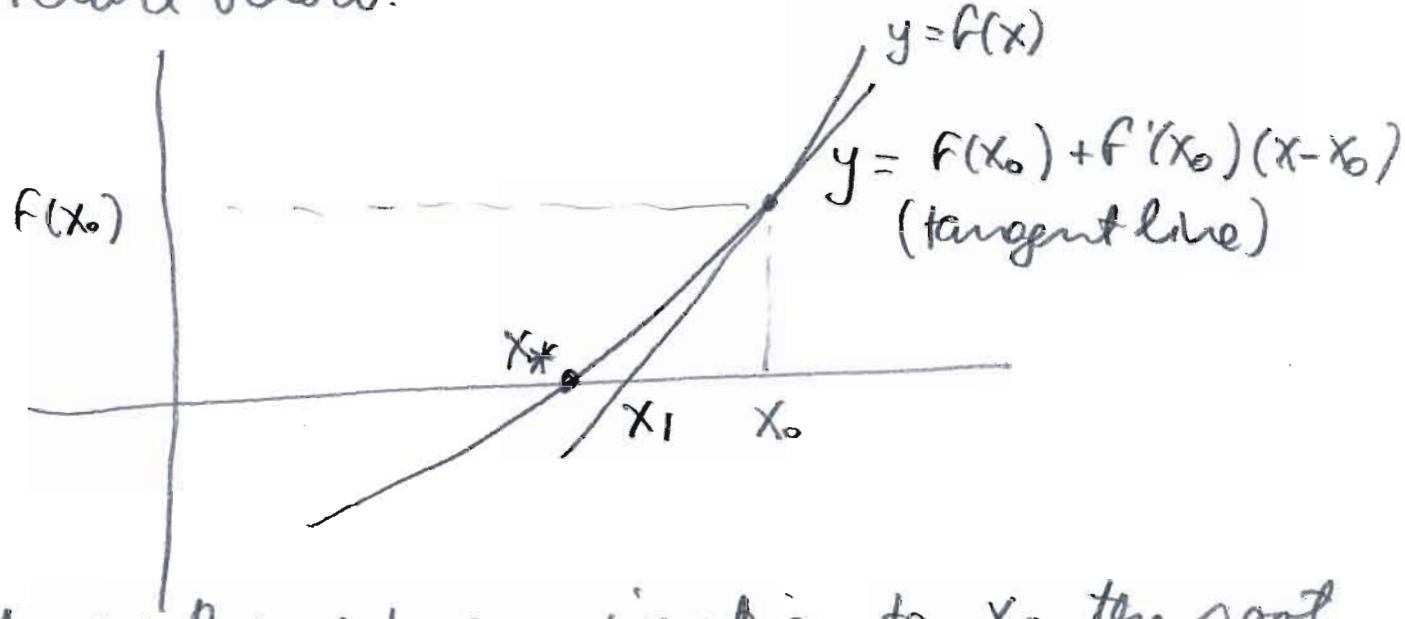
Note that if there are multiple roots in (a,b) then the method will converge to one of them.

Note that $B_{n+1} < \frac{1}{2} B_n$ ← power of 1. **Defn**

We say that bisection method converges linearly, with factor of 1/2.

Newton's Method

Begin with a relatively accurate estimate x_0 of the root. Consider the linear (tangent line) approximation of $f(x)$ based at x_0 as in the picture below.



Take as the next approximation to x_0 the root of the tangent line. That is, take x_1 that satisfies

$$f(x_0) + f'(x_0)(x_1 - x_0) = 0$$

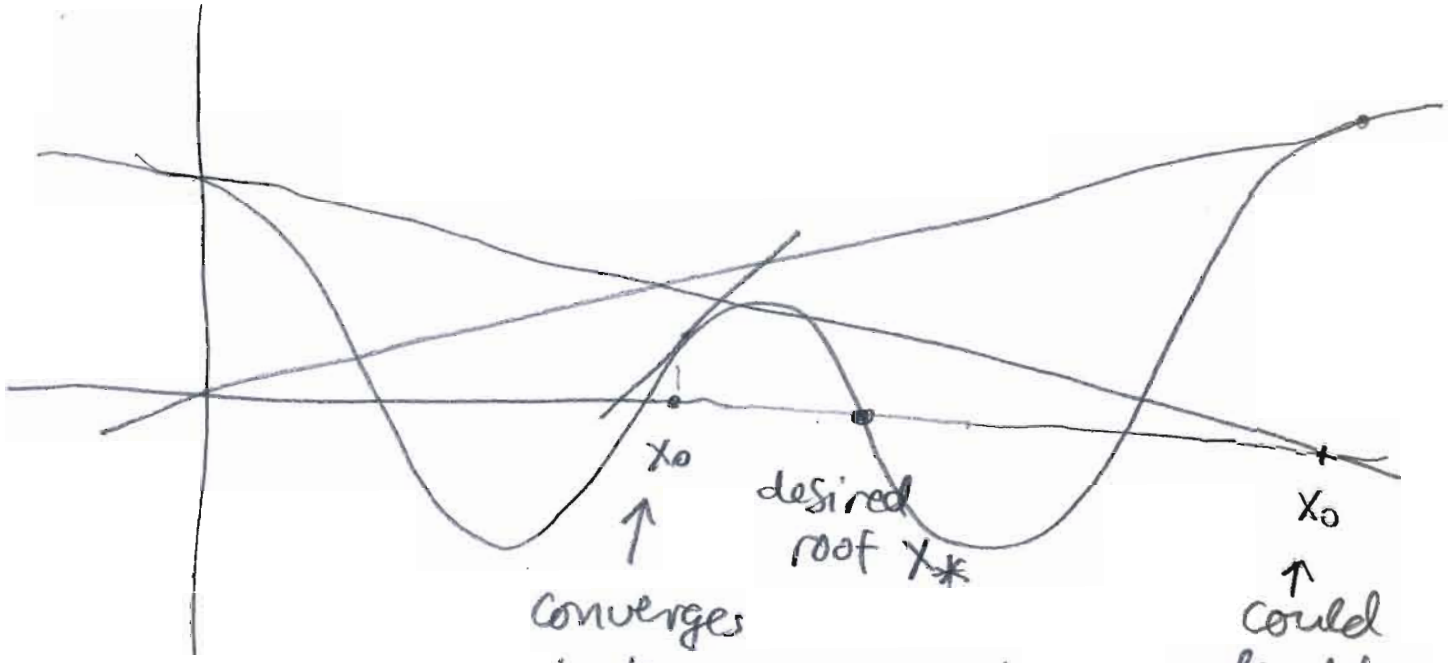
as shown in the diagram above. This can be solved for

$$x_1 = x_0 - f(x_0)/f'(x_0)$$

and can be continued iteratively.

$$x_{n+1} = x_n - f(x_n)/f'(x_n).$$

Notes: If x_0 is too far from x^*, $\{x_n\}$ may not converge to x^*, may not converge at all. We say Newton's Method is not robust. **Defn**



However, when Newton's method does converge (guaranteed when x_0 is "close enough" to x_*) it converges very quickly.

$$x_{n+1} = x_n - g(x_n), \qquad g(x) = \frac{f(x)}{f'(x)}. \tag{1}$$

Assume f is C_3 (continuous third derivatives) and that $f'(x_*) \neq 0$. Do a quadratic Taylor approximation of $g(x)$ at x_*:

$$g(x) \approx g(x_*) + g'(x_*)(x_n - x_*) + \frac{1}{2} g''(x_*)(x_n - x_*)^2 \tag{2}$$

good for x "near" x_* ; $x_n - x_*$ = error e_n ; $(x_n - x_*)^2 = e_n^2$

(More detail on Taylor approximation in next lecture).

Calculation gives

$$g(x) = 1 - \frac{f(x)f''(x)}{[f'(x)]^2} \qquad g'(x_*) = 1.$$

$$g''(x_*) = -\frac{f''(x_*)}{f'(x_*)}$$

So subtracting x_* from both sides of (1) and using (2) with the values above gives

$$e_{n+1} \approx e_n - e_n + \underbrace{\frac{f''(x_*)}{f'(x_*)}}_{c} e_n^2$$

i.e. $e_{n+1} \approx c e_n^2$.

This behaviour is known as quadratic convergence.

Defn

Note: If sufficient accuracy is required, quadratic convergence is always faster than linear convergence.

Ex

n ↓	linear $e_{n+1} \approx \frac{1}{2} e_n$.	quadratic $e_{n+1} \approx e_n^2$
0	1/2	1/2
1	1/4	1/4
2	1/8	1/16
3	1/16	1/256
	⋮	⋮

For a given accuracy tolerance δ, the number of bisection iterations $B(\delta)$ is not necessarily larger than the number of Newton iterations $N(\delta)$. However, if Newton's method converges

$$\lim_{\delta \to 0} \frac{N(\delta)}{B(\delta)} = 0.$$

Summary: Bisection is robust but converges slowly; Newton's method converges quickly but is not robust.

There is a relative of Newton's method called the secant method that does not require derivative values. There is a combination of bisection and secant called the Illinois method that is robust and converges quickly (better than linear but not quadratic).

Ex Consider finding the positive root of

$$f(x) = x^2 - 3$$

with both bisection & Newton's method.
Here $x_* = \sqrt{3}$.

Newton's iterates $x_{n+1} = x_n - \dfrac{x_n^2 - 3}{2x_n}$

$$= \frac{1}{2}\left(x_n + \frac{3}{x_n}\right)$$

Note that in this form, x_{n+1} is the average of two values, one smaller than $\sqrt{3}$ and the other larger. This formula was known to the ancient Babylonian civilization. Note that both Newton and bisection lead to convergent approximations of $\sqrt{3}$ using only arithmetic operations.

MATLAB demo programs newton1, newton2, and bisection will be shown, and the MATLAB command fzero.

Finding roots of nonlinear systems is also of interest. Let's consider the simplest case where roots (x,y) of

$$f(x,y) = 0 \quad \text{and} \quad g(x,y) = 0 \qquad (3)$$

This can be put into vector notation

$$\underline{x} = (x,y), \quad \underline{f} = (f,g)$$

So (3) becomes

$$\underline{f}(\underline{x}) = \underline{0}$$

In this form, we can consider nonlinear systems of arbitrary size N (number of components of $\underline{x}$ and $\underline{f}$, that is the number of unknowns and equations, respectively).

The $N=2$ case (1) has the graphical interpretation of (x,y) being the intersection of the zero level line of f and the zero level line of g. As for scalar nonlinear problems, (3) may have no solutions, a single solution or multiple solutions.

Newton's method can be extended to the vector case (3). Starting at a point (x_0, y_0) near a root (x^*, y^*) we can approximate f & g by their tangent planes at (x_0, y_0):

$$f(x,y) \approx f(x_0,y_0) + \frac{\partial f}{\partial x}(x_0,y_0)(x-x_0) + \frac{\partial f}{\partial y}(x_0,y_0)(y-y_0)$$

$$g(x,y) \approx g(x_0,y_0) + \frac{\partial g}{\partial x}(x_0,y_0)(x-x_0) + \frac{\partial g}{\partial y}(x_0,y_0)(y-y_0)$$

not both zero but close.

As in the scalar case (x_1, y_1) is chosen so that the tangent plane approximations are both zero, that is

$$\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} \begin{bmatrix} x_1 - x_0 \\ y_1 - y_0 \end{bmatrix} = - \begin{bmatrix} f(x_0,y_0) \\ g(x_0,y_0) \end{bmatrix} \quad (4)$$

Defn

Jacobian matrix J evaluated at (x_0, y_0).

Note that (4) is a linear system. We assume that $\mathbb{J}(\underline{x}_*)$ is invertible and so $\mathbb{J}(\underline{x}_0)$ is invertible if $\underline{x}_0$ is close enough to $\underline{x}_*$. Geometrically, this condition implies that the zero level lines of f and g are not parallel at the root. We can write (4) in vector form

$$\mathbb{J}(\underline{x}_0)(\underline{x}_1 - \underline{x}_0) = -\underline{f}(\underline{x}_0).$$

↑ in column vectors

Which can be solved for

$$\underline{x}_1 = \underline{x}_0 - \left[\mathbb{J}(\underline{x}_0)\right]^{-1} \underline{f}(\underline{x}_0).$$

and then repeated

$$\underline{x}_{n+1} = \underline{x}_n - \left[\mathbb{J}(\underline{x}_n)\right]^{-1} \underline{f}(\underline{x}_n). \qquad (5)$$

The form (5) is how it is usually written, but in practice one solves for

$$\underline{u}_n = \underline{x}_{n+1} - \underline{x}_n$$

$$\mathbb{J}_n \underline{u}_n = -\underline{f}(\underline{x}_n)$$

↑ Jacobian evaluated at $\underline{x}_n$

and then update

$$\underline{x}_{n+1} = \underline{x}_n + \underline{u}_n .$$

Note: An important example of nonlinear systems comes in the optimization (maximization or minimization) of functions of many variables

$$f(\underline{x})$$

↑

scalar F.

Optimized at critical points $\nabla f = \underline{0}$. Here, the Jacobian of ∇f is the Hessian Matrix of F:

$$\mathbb{J}_{ij} = \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_i}\right) = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

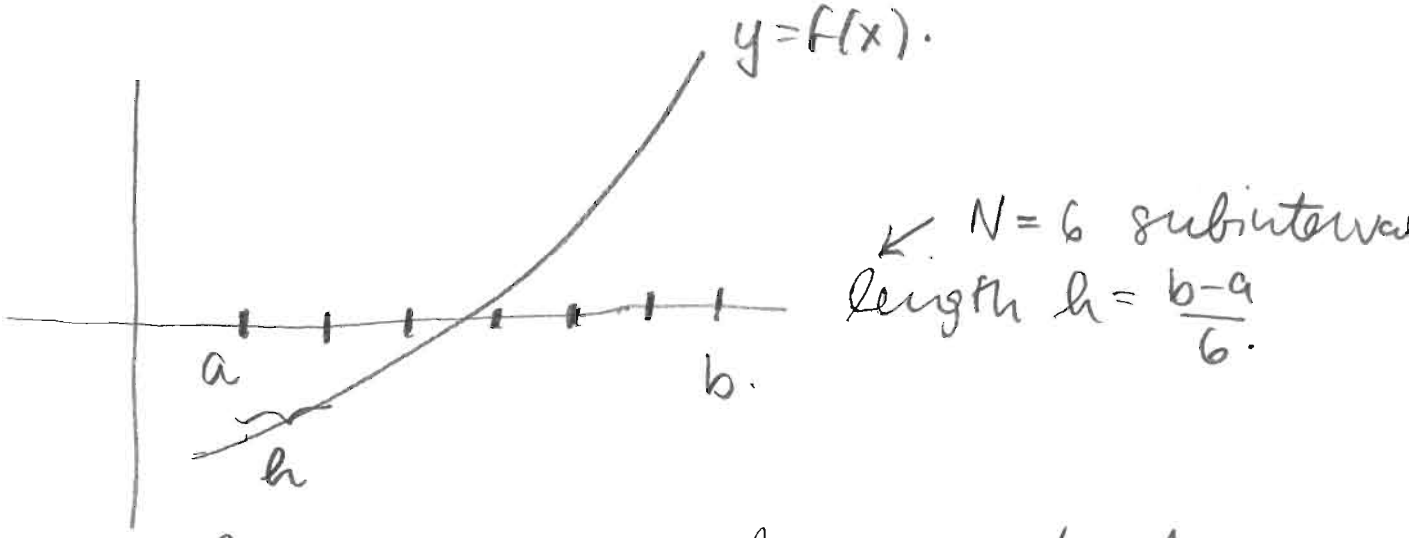
You can learn more about optimization in CS 406 next term...

Refs: $\mathbb{J}$ can be approximated using numerical differentiation and there are techniques to improve efficiency with fast, approximate updates for $\mathbb{J}_n^{-1}$ (Broyden's Method)

Math 405, Fall 2014, Day 3 notes.

A main goal of the course is to approximate solutions to DE's. As a start, let's try to approximate continuous scalar functions using a finite number of values.

A typical set-up would be considering $f(x)$ on an interval $[a,b]$. We divide the interval into N subintervals of equal length $h = \frac{b-a}{N}$. On each subinterval, we approximate $f(x)$ with a polynomial, which is specified with a finite number of values.



Suppose for every N we have a strategy for a piecewise polynomial approximating function $A_N(x)$. We say that the strategy converges (in maximum norm)

if $\lim_{N\to\infty} \|f - A_N\|_\infty = 0$.
(equivalently $h \to 0$)

where $\|f\|_\infty = \max_{a \le x \le b} |f(x)|$.

If $\|f - A_N\|_\infty \le C h^q$ with $q \ge 0$ we say the method converges with order q. As with iterative methods, approximation methods with higher order convergence always outperform lower order methods if the accuracy required is high enough.

linear (Tangent line) approximation

approximating $f(x)$ near $x=a$ by its tangent line approximation

$$f(x) \approx f(a) + f'(a)(x-a) := L(x)$$

How accurate is this approximation? (assume $f \in C_2$)

Theorem $f(x) - L(x) = \dfrac{f''(\xi)}{2}(x-a)^2$ } error.

for some ξ between x and a.

This expression makes sense:

- the approximation can get worse (the error gets larger) as $|x-a|$ gets larger.

- If F'' is large near a, then the approximation may not be accurate unless $|x-a|$ is very small.
- If $F''(x) \geq 0$ (F is concave up) for x near a then the linear approximation will underestimate the function values.

Proof: uses Rolle's theorem as a lemma:

Rolle: If F is differentiable and $F(a)=0$ and $F(b)=0$ then there is a point ξ in (a,b) at which $F'(\xi)=0$.

Pick a point x (call it $x=b$) at which the approximation is made. That is, we want to prove the statement of the theorem for $x=b$.

Consider $q(x) = F(x) - L(x) - K(x-a)^2$ where K is the constant

$$K = \frac{F(b)-L(b)}{(b-a)^2}.$$

Note that $q(a)=0$ and $q(b)=0$ so (Rolle!) $q'(\xi_1)=0$ for some $\xi_1 \in (a,b)$.

Compute

$$q'(x) = F'(x) - L'(x) - 2K(x-a).$$

Since $L'(x) = F'(a)$,

$$q'(a) = 0.$$

Since $q'(a)=0$ and $q'(\xi_1) = 0$ (Rolle) $q''(\xi) =$
for some ξ in $(a, \xi_1) \subset (a,b)$. Compute

$$q''(x) = F''(x) - 2K$$

Thus $q''(\xi) = 0 \Rightarrow F''(\xi) = 2\dfrac{F(b)-L(b)}{(b-a)^2}$

which proves the result.

Let's apply the result to our finchian approximation on subintervals. Consider

$$\max_{x\in[a,b]} |F''(x)| := K_2.$$

If we use linear approximation on subintervals the best we can do is place the base points at subinterval centers. The resulting $A_N(x)$ has

$$\|f - A_N\|_\infty \le \frac{K_2}{2}\left(\frac{h}{2}\right)^2 = K_2 h^2/8$$

and requires $2N$ data (one value of f and one of f' on every subinterval). This is a second order approximation.

Note: $A_N(x)$ is not a continuous function.

We can also use higher order Taylor polynomial approximation

quadratic $Q(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2$

general order $T_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i$

Notation $f^{(i)}$, i'th derivative of f, convention $0! = 1$, $0^\circ = 1$, $f^{(0)} = f$.

Theorem $f(x) - T_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$

for some $\xi \in (a, x)$.

If $f \in C^{n+1}$ we can use piecewise $T_n(x)$ approximation to get an $(n+1)$th order approx

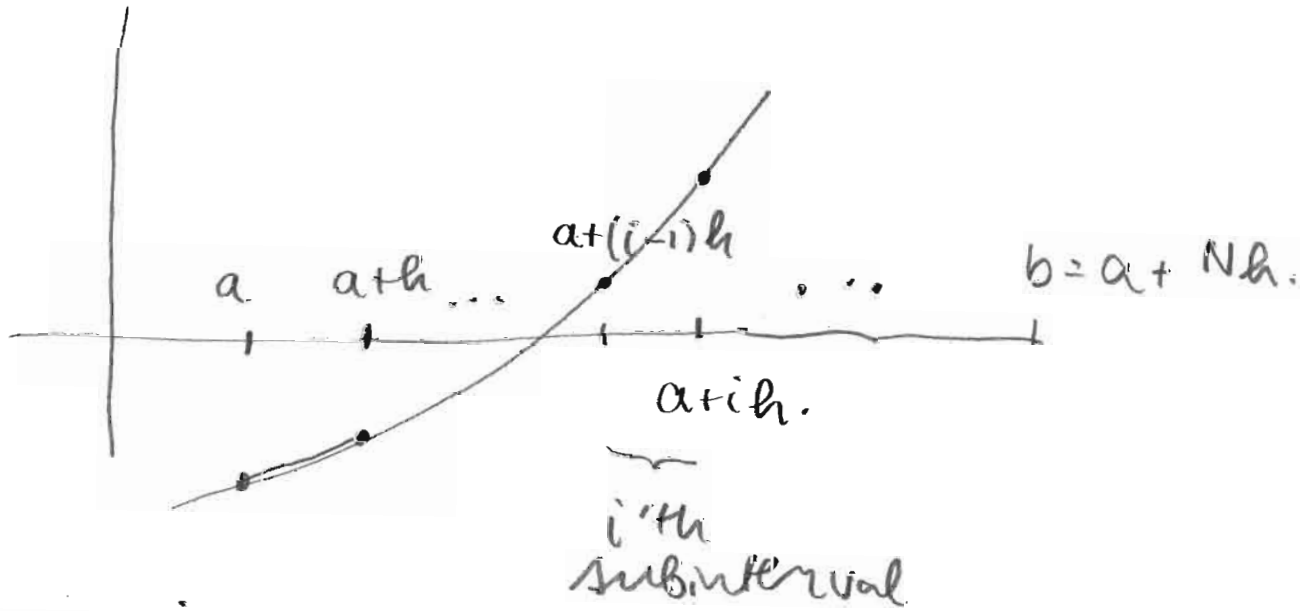
of f.

linear Interpolation

Use the N+1 values of f at subinterval ends to build an approximation using line segments joining these points.

Advantages over tangent line approximation:

- don't need derivative values
- resulting approximating function is continuous.
- more accurate (second order but smaller constant).



On the i'th subinterval, f(x) is approximated by

$$I(x) = f(a+(i-1)h)(a+ih-x)/h + f(a+ih)(x-(a+(i-1)h))/h$$

Note: This has the form

$$I(x) = f_{i-1}\,\varphi_1(x) + f_i\,\varphi_2(x)$$

shape functions - polynomials that have value 1 for one function value and zero for the others.

Good idea: Map each subinterval onto a nice, reference interval: either $[0,1]$ or $[-1,1]$.

↑ preferred if there is some even/odd function simplification possible.

$$x \in [a+(i-1)h,\ a+ih] \longmapsto y \in [0,1]$$

$$y(x) = (x - (a+(i-1)h))/h.$$

Now $I(x)$ can be written

$$I(y) = f(0)\,(1-y) + f(1)\,y$$

$f(y)$: data values

reference shape functions

We can do the error analysis in the "y" setting - the algebra is much easier.

Theorem $f(y) - I(y) = \frac{1}{2}\, y(1-y)\, \frac{d^2f}{dy^2}(\xi)$

for $\xi \in (0,1)$.

$y(1-y)$: maximum value $1/4$ on $y \in [0,1]$.

Note that $\frac{df}{dy} = \frac{df}{dx} \cdot \frac{dx}{dy} = \frac{df}{dx} \Big/ \frac{dy}{dx} = h \frac{df}{dx}$.

and $\frac{d^2f}{dy^2} = h^2 \frac{d^2f}{dx^2}$.

So if we return to the original interval,

$$|f(x) - I(x)| < \frac{1}{8} K_2 h^2.$$

Q: This is the same second order error as Tangent line approx. Why did I say that linear interpolation is more accurate? (class discussion).

quadratic interpolation

Suppose we had function values at subinterval ends and centres. We could

then do quadratic interpolation. We could consider this on the reference interval $y \in [-1,1]$:

$$f(y) \approx S(y) := f(-1)\,\frac{y(y-1)}{2} + f(0)\,(1-y^2) + f(1)\,\frac{y(y+1)}{2}$$

Note: Same form: data values times shape functions.

Theorem: $f(y) - S(y) = \frac{1}{6}\, y(y^2-1)\,\frac{d^3 f}{dy^3}(\xi)$ for $\xi \in (-1,1)$.

When applied to subintervals of size h, the approximation has third order accuracy.

Other function approximations:

Cubic Hermite Interpolation

Reference interval $y \in [0,1]$. Data $f(0), f(1), f'(0), f'(1)$ given.

$$f(y) \approx C(y) := f(0)(2y^3 - 3y^2 + 1) + f(1)(-2y^3 + 3y^2) + f'(0)(y^3 - 2y^2 + y) + f'(1)(y^3 - y^2)$$

Theorem: $F(y) - C(y) = \frac{1}{24} y^2(y-1)^2 \frac{d^4 f}{dy^4}(\xi)$.

So will be fourth order accurate when applied to subintervals of size h.

Math 405, Fall 2014, Day 4 Notes

Note: In the function approximation so far, we have used fixed order polynomial approximation on subintervals that $\to 0$ in size.

We could also consider polynomials on the whole interval of increasing order. Matching high order polynomials to equally spaced function values does not in general lead to a convergent scheme. (here convergence is in the polynomial order $M \to \infty$). It is possible to make this idea work using function values at unequally spaced points. One technique is called Chebyshev interpolation.

Chebyshev interpolation is a bit too tricky for us to begin with, so let us start with an easier but related idea: Fourier Interpolation of periodic functions.

Assume $f(x)$ is C_m (2π-periodic). Fourier coefficients of f are

$$\hat{f}_\alpha = \frac{1}{2\pi} \int_0^{2\pi} f(x)\, e^{-i\alpha x}\, dx. \qquad \alpha \in \mathbb{Z}$$

$$\cos(\alpha x) - i \sin(\alpha x).$$

Note: This is the complex form of the Fourier series, but the same story as the real version if that is all you have seen.

If you knew the $\hat{f}_\alpha$ values you can recover $f(x)$ values from

$$f(x) = \sum_{\alpha=-\infty}^{\infty} \hat{f}_\alpha\, e^{i\alpha x} \qquad (1).$$

Notes: • The 2π factor and the placement of the +/- in the complex exponentials can vary in the literature.

• Fourier analysis is essentially the representation of functions by their frequency components.

We make a Fourier Interpolant by using a finite number of terms in (1):

$$f(x) \approx F_M(x) := \sum_{\alpha=-M}^{M} \hat{f}_\alpha e^{i\alpha x} \qquad (2)$$

Q: How accurate is this approximation? (assume for now that $\hat{f}_\alpha$ values are known exactly).

$$f(x) - F_M(x) = \sum_{|\alpha| \geq M+1} \hat{f}_\alpha e^{i\alpha x}$$

So $\|f - F_M\|_\infty \leq \sum_{|\alpha| \geq M+1} |\hat{f}_\alpha|.$

From this, we can see that the approximation will be good if the size of the Fourier coefficients decays rapidly as $|\alpha| \to \infty$.

$$\hat{f}_\alpha = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-i\alpha x} \, dx$$

↘ exclude $\alpha = 0$, integrate by parts

$$= -\frac{1}{2\pi} \int_0^{2\pi} \frac{f'(x) e^{-i\alpha x} \, dx}{i\alpha}.$$

The boundary terms from integration by parts cancel out since all functions are periodic.

Since $f \in C_m$ we can integrate by parts $m-1$ more times to get

$$\hat{f}_\alpha = \frac{(-1)^m}{2\pi} \int_0^{2\pi} \frac{f^{(m)}(x) e^{-i\alpha x}}{(i\alpha)^m} \, dx$$

since $|e^{i\alpha x}| = 1$ and using

$$K_m = \max_{0 \le x \le 2\pi} |f^{(m)}(x)|.$$

we have

$$|\hat{f}_\alpha| \le \frac{K_m}{|\alpha|^m}$$

So $$\|f - F_M\|_\infty \le 2K_m \sum_{\alpha = M+1}^{\infty} \frac{1}{\alpha^m} \qquad (3)$$

↑ accounts for $\pm\alpha$ values.

In what follows, suppose $m > 2$. The following estimate may not be optimal [discuss in class].

In (3) write $\frac{1}{\alpha^m} = \frac{1}{\alpha^2} \frac{1}{\alpha^{m-2}} \le \frac{1}{\alpha^2} \frac{1}{(M+1)^{m-2}}$ in the sum. Thus,

$$\|f - F_M\|_\infty \le \frac{2K_m}{(M+1)^{m-2}} \sum_{\alpha=M+1}^{\infty} \frac{1}{\alpha^2} \le \frac{2K_m}{(M+1)^{m-2}} \underbrace{\sum_{\alpha=1}^{\infty} \frac{1}{\alpha^2}}_{\pi^2/6}$$

So, $\|f - F_M\|_\infty \leq \frac{\pi^2}{3} K_m / (M+1)^{m-2}$. (*).

Note that (*) is not just one estimate, but many different ones for every m up to the maximum differentiability of the function. We see that convergence (in $M \to \infty$) is of order up to $m-2$. Thus, the rate of convergence is not limited by the approx method, only the smoothness of the function. This type of convergence is called spectral convergence Important Defn.

So far, the $\hat{f}_\alpha$ values are determined by periodic integrals - at the top of page 2. These are integrals with highly oscillatory integrands - hard to evaluate numerically. We will follow an idea below that will allow us to easily approximate the $\hat{f}_\alpha$ values in a way that retains spectral convergence of the truncated Fourier approximation.

Consider (2) evaluated at an equally spaced grid of $2M$ points, $\{x_j\}$, where

$x_j = jh$ and $h = \pi/M$.

That is

$$f(x_j) \approx F_M(\pi j/M) = \sum_{\alpha=-M}^{M} \hat{f}_\alpha e^{i\alpha j\pi/M}. \quad (4)$$

Note that if $\alpha = -M$,

$$e^{i\alpha j\pi/M} = e^{-ij\pi} = (-1)^j$$

also if $\alpha = +M$

$$e^{i\alpha j\pi/M} = e^{+ij\pi} = (-1)^j$$

Thus, on the grid contributions from $\hat{f}_M$ and $\hat{f}_{-M}$ can't be distinguished. This is a consequence of a phenomenon called aliasing: More generally, $\hat{f}_{\alpha+2M}$ can't be distinguished from $\hat{f}_\alpha$ on this grid of $2M$ points. We can replace (4) by

$$f(x_j) \approx \sum_{\beta=0}^{2M-1} g_\beta e^{i\beta j\pi/M} \quad (5)$$

where $g_\beta = \begin{cases} \hat{f}_\beta & 0 \le \beta < M. \\ \hat{f}_M + \hat{f}_{-M} & \beta = M. \\ \hat{f}_{\beta-2M} & M < \beta \le 2M-1. \end{cases}$

Note that working out all $2M$ values $j = 0, \ldots 2M-1$ of (5) is equivalent to multiplying by a $2M \times 2M$ matrix:

$$\underline{F} \approx \mathbb{F}\, \underline{g} \qquad (6).$$

$\underline{F}$: column vector of approximate $f(x_j)$ values

$\underline{g}$: column vector of g_β values.

$$F_{j\beta} = e^{i\beta j \pi/M}.$$

Multiplication as written would take $4M^2$ operations but it can be done much faster using the Fast Fourier Transform. Details are in the attached notes.

In addition, we can use (6) backwards

$$\underline{g} \approx \mathbb{F}^{-1}\, \underline{F}$$

$\underline{g}$: approximate, aliased $\hat{F}_\alpha$ values.

$\underline{F}$: exact, known values of f on the grid points.

Multiplication by F^{-1} can also be done quickly using the inverse FFT.

It can be shown that approximating $\hat{f}_\alpha$ values in this way retains spectral accuracy of the approximation (2).

Notes:
- The ordering of g_β coefficients at the bottom of page 6 is typical for FFT implementations.
- If $f(x)$ is a real function, $\hat{f}_\alpha$ and $\hat{f}_{-\alpha}$ will be conjugates. Thus, there is only $2M$ "real" information in g. There are FFT implementations that take advantage of this. You will need to look at the details to determine what exactly is output from these FFT routines.

We can describe algorithms or problems of size M with operation counts $\leq CM^p$ as $O(M^p)$. Multiplying by F is $O(M^2)$ but evaluating using the FFT is $O(M \log M)$. [much faster]. This assumes M is a power of 2 or has many small prime factors.

Math 405, Fall 2014, Day 5

In Day 3 notes, we represented $f(x)$, $x \in [a,b]$ approximately using piecewise polynomial functions

$$f(x) \approx A_N(x)$$

[N number of subintervals, length $h = (b-a)/N$].

Today, we are going to see how to approx

$$J := \int_a^b f(x)\,dx.$$

A natural, first idea is to take

$$J \approx \int_a^b A_N(x)\,dx$$

polynomials on sub-intervals, integration can be done exactly ✓

This is a good place to start, but there are also some other techniques to get good approximations to integrals (numerical quadrature).

Defn

Trapezoidal Rule

Take $A_N(x)$ to be linear interpolation on the subintervals $[x_{j-1}, x_j]$, where

$$x_j = a + jh$$

(so $x_0 = a$, $x_N = b$).

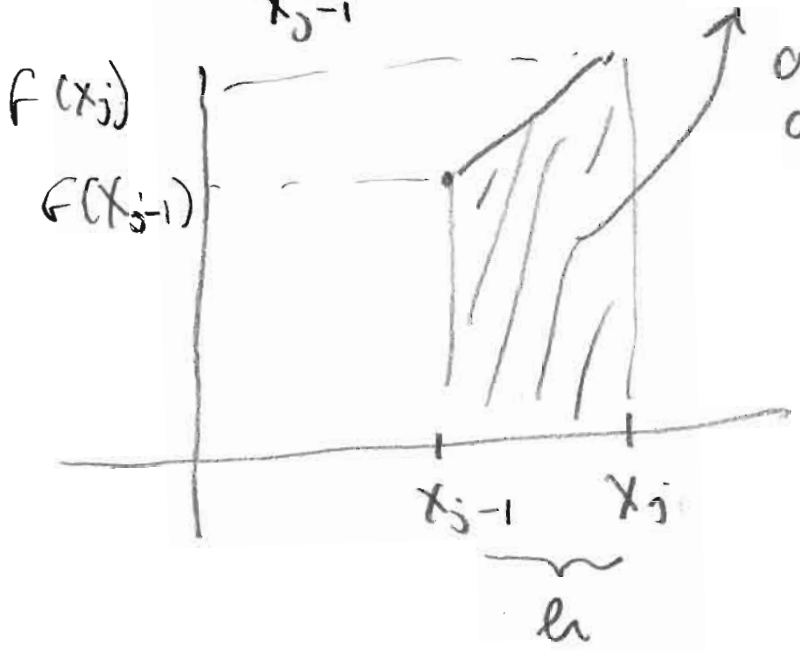
$$J = \int_a^b f(x)dx = \sum_{j=1}^{N} \int_{x_{j-1}}^{x_j} f(x)dx \approx \sum_{j=1}^{N} \int_{x_{j-1}}^{x_j} I_j(x)dx \quad (1)$$

still exact, just dividing the integration into subintervals

now approximate

linear interpolation

Consider $\int_{x_{j-1}}^{x_j} I_j(x)dx = \frac{h}{2}\left(f(x_{j-1}) + f(x_j)\right)$.



area of trapezoid.

Note: The approximation to the integral on a subinterval is a linear combination of function values. This is the case for all quadrature methods.

Summing over all subintervals, we get

$$J \approx \frac{h}{2}\left(f(a) + 2f(x_1) + 2f(x_2) + \cdots 2f(x_{N-1}) + f(b\right.$$

interior points get contributions from subintervals on both sides

This defines the Trapezoidal Rule.

Q: How accurate is this method?

Consider the error in one subinterval

$$E_j = \int_{x_{j-1}}^{x_j} \left(F(x) - I_j(x) \right) dx$$

Map the interval onto the reference interval $y \in [0,1]$.

$$y = (x - x_{j-1})/h \,, \quad \frac{dy}{dx} = \frac{1}{h} \,, \quad \text{"} dx = h\,dy \text{"}$$

↙ substitution in the integral.

$$E_j = h \int_0^1 \underbrace{(F(y) - I(y))}\, dy$$

↘ use Day 3 ideas.

$$\frac{1}{2} y(1-y) \frac{d^2 f}{dy^2}(\xi)$$

$$\left| \frac{d^2 f}{dy^2}(\xi) \right| \le h^2 K_2$$

$$|E_j| \le h \int_0^1 |F(y) - I(y)|\, dy$$

$$\le \frac{h^3 K_2}{2} \underbrace{\int_0^1 y(1-y)\, dy}_{1/6}$$

← integrand is positive no need for absolute values.

$$= \frac{h^3 K_2}{12}$$

Error in I from Trapezoidal Rule approximation

Now $$\left| I - \int_a^b A_N(x)\,dx \right| = \left| \sum_{j=1}^{N} E_j \right|$$

$$\leq \sum_{j=1}^{N} |E_j| \leq \frac{N h^3 K_2}{12} = \frac{b-a}{12} K_2 h^2.$$

($N = \frac{b-a}{h}$)

Thus, we have proved the second order convergence of the Trapezoidal Rule approximation.

General Idea

Consider quadrature methods on the reference interval $[-1,1]$. The general form is

$$\int_{-1}^{1} f(y)\,dy \approx \sum_{m=1}^{M} w_m f(y_m) \qquad (2)$$

M points with weights w_m at positions y_m specify the method.

Ex Trapezoidal Rule $M=2$, $w_1 = w_2 = 1$, $y_1 = -1$, $y_2 = +1$.

Theorem: If (2) is exact for polynomials $f(y)$ up to order q, but not exact for all polynomials of order $q+1$, then (2) is an order $q+1$ quadrature method when applied

to subintervals of length h.

Proof: Expand $f(y)$ in a Taylor series of order q based at $y=0$.

$$f(y) = T_q(y) + \frac{f^{(q+1)}(\xi)}{(q+1)!} y^{q+1}$$

↙ different ξ for every y.

Consider the error in (2)

$$\left| \int_{-1}^{1} f(y)dy - \sum_{m=1}^{M} w_m f(y_m) \right|$$

$$\leq \left| \int_{-1}^{1} T_q(y)dy - \sum_{m=1}^{M} w_m T_q(y_m) \right| + \cdots$$

↙ zero, since by assumption the quadrature rule is exact for polynomials of this order.

$$\left| \int_{-1}^{1} \frac{f^{(q+1)}(\xi)}{(q+1)!} y^{q+1} dy - \sum_{m=1}^{M} w_m \frac{f^{(q+1)}(\xi)}{(q+1)!} y_m^{q+1} \right| \cdots$$

all $|f^{(q+1)}(\xi)|$ terms are $\leq \left(\frac{h}{2}\right)^{q+1} K_{q+1}$, so

$$\cdots \leq \frac{h^{q+1} K_{q+1}}{2^{q+1}(q+1)!} \left\{ 2 \int_{0}^{1} y^{q+1} dy + \sum_{m=1}^{M} |w_m y_m^{q+1}| \right\}.$$

↖ This step may give a non-optimal constant.

$$\leq C h^{q+1}.$$

As in the analysis of the trapezoidal rule on page 3, the integral on subintervals is scaled by $h/2$ and then there are $N = (b-a)/h$

subintervals to sum over. The total error in approximating

$$\int_a^b f(x)\,dx$$

is then $\leq \frac{(b-a)}{2} C h^{q+1}$. Thus, we have $q+1$ order convergence as stated in the theorem.

Simpson's Rule

$$\int_{-1}^{1} 1\,dy = 2$$

$$\int_{-1}^{1} y\,dy = 0.$$

$$\int_{-1}^{1} y^2\,dy = 2/3$$

$$\int_{-1}^{1} y^3\,dy = 0$$

$$\int_{-1}^{1} y^4\,dy = 2/5.$$

$M=3$, $W_1 = 1/3$, $y_1 = -1$

$W_2 = 4/3$, $y_2 = 0$

$W_3 = 1/3$ $y_3 = 1$.

$$\sum_{m=1}^{3} W_m f(y_m)$$

exact for polynomials up to third order.

Thus, Simpson's rule is 4th order accurate.

M=2 Gaussian quadrature

$M=2$, $w_1 = w_2 = 1$, $y_{1,2} = \pm\sqrt{1/3}$.

Math 405, Fall 2014, Day 6

We test Trapezoidal Rule & Simpson's rule on a known integral

$$J = \int_0^1 \sin x\,dx = 1 - \cos(1) \approx 0.4596977$$

The codes are posted. Known error bounds for the approximations of $\int_a^b f(x)dx$ are

$$|J - \text{Trap}| \le \frac{b-a}{12} K_2 h^2$$

and $$|J - \text{Simp}| \le \frac{b-a}{180} K_4 h^4$$

↑ Note: this is the h between function evaluations.

$K_j = \max_{a \le x \le b} |f^{(j)}(x)|$ ↑ f is the integrand.

We observe that these bounds are obeyed by our approximations (no surprise, these have been proved). However, convergence has a better character than expected from the expressions above. We observe

$$J \approx \text{Trap} + C h^2$$

given constant $C \le \frac{b-a}{12} K_2$.

It can be shown that

$$J = \text{Trap} + C h^2 + \underbrace{O(h^4)}$$

with

$$C = -\frac{b-a}{12}\left[f''\right]_{\text{ave}}.$$

Notation $O(h^4)$ is a quantity smaller than a constant times h^4 as $h \to 0$ in absolute value.

If you have

$$J = \text{Trap}(h) + Ch^2 + O(h^4) \quad \text{for all } h,$$

then $J = \text{Trap}(h/2) + \frac{Ch^2}{4} + O(h^4)$.

and you can combine these results (Richardson extrapolation)

$$J = \frac{4}{3}\text{Trap}(h/2) - \frac{1}{3}\text{Trap}(h) + O(h^4).$$

chosen to cancel out Ch^2 error.

Note that $\frac{4}{3}\text{Trap}(h/2) - \frac{1}{3}\text{Trap}(h)$ is the same as $\text{Simp}(h/2)$, so this is another way to derive Simpson's rule.

Adaptive Methods

The error in local Trapezoidal rule is $\frac{h^3}{12} f''(\xi)$. It makes sense to make h smaller in regions where f'' is large. We can do this adaptively, estimating how large the error is for a given interval using the regularity in the error as the grid is refined.

Specify an error tolerance δ for

$$J = \int_a^b f(x)\,dx.$$

If we can make the error smaller than

$\frac{\delta h}{(b-a)}$ on subintervals of length h, we can guarantee accuracy of δ for the integral on the whole interval.

Start with N uniform subintervals of length $h=(b-a)/N$ and consider Trapezoidal rule approximation on one of them. The error is

$$E_h = \frac{f''(\xi)\,h^3}{6} \approx \frac{f''(x_*)\,h^3}{6} \qquad (1).$$

where ξ is somewhere in the subinterval and x_* is the centre of the subinterval. Now divide the subinterval into 2 subintervals of length $h/2$. This has error (over both smaller intervals).

$$E_{h/2} = \frac{f''(\xi_1)}{6}\left(\frac{h}{2}\right)^3 + \frac{f''(\xi_2)}{6}\left(\frac{h}{2}\right)^3$$

where ξ_1 is in the left half subinterval and ξ_2 is in the right half subinterval.

$$E_{h/2} \approx \frac{f''(x_*)}{6} h^3 \cdot 1/4. \qquad (2).$$

Comparing (1) & (2) we see that

$$E_{h/2} \approx (E_h - E_{h/2})/3. \leftarrow \text{error estimator.}$$

So we accept the accuracy of the two $h/2$ Trapezoid Rule approximations on this interval

of length h if

$$\frac{(E_h - E_{h/2})}{3} \leq \frac{\delta h}{(b-a)} \qquad (3)$$

↑ estimated error on an interval of length h

↖ allowable error.

In practice, a "safety" factor $\theta < 1$ is typically used

$$\frac{(E_h - E_{h/2})}{3} \leq \theta \frac{\delta h}{(b-a)}.$$

since (1) & (2) are only approximations. If the tolerance (3) is not satisfied, then both $h/2$ subintervals are divided into two subintervals of length $h/4$ and the process is repeated. Since the LHS of (3) is $O(h^3)$ and the RHS is $O(a)$, this process will terminate if f is C_2. In practice, a maximum refinement level is specified, and the method reports a failure if tolerance is not reached at that level.

In the approach above, grid refinement is used to estimate the error. Another approach is to use a higher order method to estimate the error. For example, we can take Gaussian quadrature of order 4 and Gaussian quadrature of order 6.

On a subinterval of length h

G_4 error $O(h^5)$

G_6 error $O(h^7)$

If h is sufficiently small, then

$|G_4 - G_6| \approx G_4$ error.

Note: It is more computationally efficient to use a higher order quadrature method using the function values you have already computed (Gauss-Kronrod rules).

Note: adaptivity can also be in increasing the order of approximation in a subinterval. Grid refinement is called "h-adaptivity" and order increase is "p-adaptivity". They could both be used "hp-adaptivity".

Math 405, Fall 2015, Day 7.

Consider now approximation of derivatives. As with integration, we can start by using piecewise polynomial approximations

$$f(x) \approx A_N(x)$$

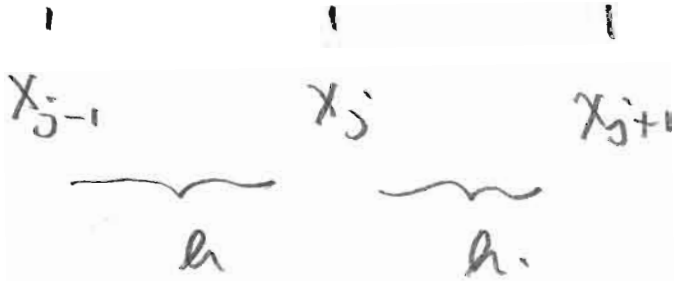
and use these to approximate derivatives

$$f'(x) \approx A_N'(x)$$

$$f''(x) \approx A_N''(x)$$

etc.

Let's consider piecewise quadratic approximation



Note that this is length $2h$

Mapped on to the reference interval $y \in [-1, 1]$.

$$y = \frac{x - x_j}{h} \qquad \frac{dy}{dx} = \frac{1}{h}$$

$$f(y) \approx Q(y) = \frac{1}{2} f(-1)\, y(y-1) + f(0)\,(1-y^2) + \frac{1}{2} f(1)\, y(y+1).$$

where $f(-1) = F(x_{j-1})$

$y=-1$ $\quad$ x

$f(0) = F(x_j)$ $\quad$ ($y=0$)

$f(1) = F(x_{j+1})$ $\quad$ ($y=1$)

We take $f'(y) \approx Q'(y) = \frac{1}{2} f(-1)(2y-1)$

$$-2y\, f(0) + \tfrac{1}{2} f(1)(2y+1). \qquad (1)$$

Usual Q: How accurate is this approximation? Let's do the work in y, then scale back to x.

Consider the approximation at $y=b$. Expand the values $f(0)$, $f(-1)$, $f(1)$ in Taylor series around b.

$$f(0) = f(b) + f'(b)(-b) + \tfrac{1}{2} f''(b) b^2 - \tfrac{1}{6} f'''(b) b^3 + \tfrac{1}{24} f''''(\xi_1) b^4.$$

$$f(1) = f(b) + f'(b)(1-b) + \tfrac{1}{2} f''(b)(1-b)^2 + \tfrac{1}{6} f'''(b)(1-b)^3 + \tfrac{1}{24} f''''(\xi_2)(1-b)^4$$

$$f(-1) = f(b) + f'(b)(-1-b) + \tfrac{1}{2} f''(b)(1+b)^2 - \tfrac{1}{6} f'''(b)(1+b)^3 + \tfrac{1}{24} f''''(\xi_3)(b+1)^4$$

Now collect terms in (1):

$$f'(b) \times Q'(b) =$$

$$F(b)\left[\tfrac{1}{2}(2b-1) - 2b + \tfrac{1}{2}(2b+1)\right] + \quad (0)$$

$$f'(b)\left[-\tfrac{1}{2}(2b-1)(b+1) + 2b^2 + \tfrac{1}{2}(2b+1)(1-b)\right] + \quad (1)$$

$$-b^2 - b/2 + 1/2 \qquad -b^2 + b/2 + 1/2$$

$$f''(b)\left[\tfrac{1}{4}(2b-1)(1+b)^2 - b^3 + \tfrac{1}{4}(2b+1)(1-b)^2\right] + \quad (0)$$

$$\tfrac{1}{2}b^3 + \tfrac{3}{4}b^2 - \tfrac{1}{4} \qquad \tfrac{1}{2}b^3 - \tfrac{3}{4}b^2 + \tfrac{1}{4}$$

$$f'''(b)\left[-\tfrac{1}{12}(2b-1)(1+b)^3 + \tfrac{1}{3}b^4 + \tfrac{1}{12}(2b+1)(1-b)^3\right]$$

$$+ O\left(\frac{d^4f}{dy^4}\right) \qquad -\tfrac{1}{6}b^4 - \tfrac{5b^3}{12} - \tfrac{1}{4}b^2 + \tfrac{b}{12} + \tfrac{1}{12}$$

$$\tfrac{1}{12} - \tfrac{b}{12} - \tfrac{1}{4}b^2 + \tfrac{5b^3}{12} - \tfrac{1}{6}b^4$$

Thus,

$$f'(b) \approx Q'(b) = f'(b) + \frac{f'''(b)}{6}(1-3b^2) + O\left(\frac{d^4f}{dy^4}\right). \quad (2)$$

Now we scale back to x_j recalling

$$\frac{d}{dx} = \frac{d}{dy}\frac{dy}{dx} = \frac{1}{h}\frac{d}{dy}$$

or $$\frac{d}{dy} = h\frac{d}{dx}.$$

Now (2) reads for $x \in [x_{j-1}, x_{j+1}]$,

$$h\frac{df}{dx}(x) - h\frac{dQ}{dx}(x) = \frac{h^3}{6}\frac{d^3f(x)}{dx^3}(1-3b^2) + O(h^4)$$

Thus,

$$\frac{df}{dx} - \frac{dQ}{dx} = \frac{h^2}{6}\frac{d^3f}{dx^3}(1-3b^2) + O(h^3)$$

dominant error term vanishes at $b = \pm 1/\sqrt{3}$ in the interval.

We have shown second order accuracy of the approximation (third order at the special points above).

If we consider the derivative approximation at grid points we have

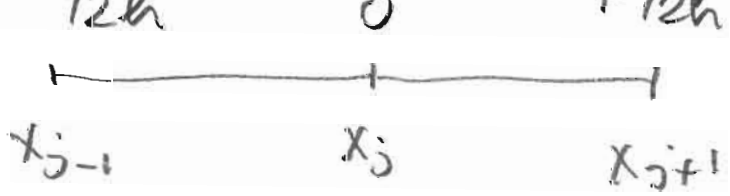
$$f'(x_{j-1}) \approx \frac{-3/2\, f(x_{j-1}) + 2f(x_j) - 1/2\, f(x_j)}{h}.$$

$$f'(x_{j+1}) \approx \frac{3/2\, f(x_{j+1}) - 2f(x_j) + 1/2\, f(x_{j-1})}{h}$$

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_{j-1})}{2h}$$

All approximations above are second order accurate. The last one is called centered differencing, the first two are second order one sided differencing.

We can represent centered differencing with the following picture, called a finite difference stencil.



Proceeding as above, we can show that

$$\frac{d^2Q}{dy^2}(y) = f(-1) - 2f(0) + f(+1). \qquad (3)$$

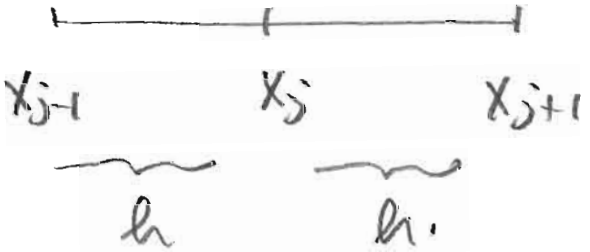
This gives the centered difference approximation of the second derivative

$$f''(x_j) \approx \frac{f(x_{j-1}) - 2f(x_j) + f(x_{j+1})}{h^2} \qquad (4)$$

This is also second order accurate although (3) is only first order accurate at other y points.

Alternate derivation of (4)

Consider



Expand $f(x_{j-1})$ and $f(x_{j+1})$ in a Taylor series about x_j:

$$f(x_{j-1}) = f(x_j) - f'(x_j)h + \frac{f''(x_j)}{2}h^2 - \frac{f'''(x_j)}{6}h^3 + \frac{f^{(4)}(\xi_1)}{24}h^4$$

$$f(x_{j+1}) = f(x_j) + f'(x_j)h + \frac{f''(x_j)}{2}h^2 + \frac{f'''(x_j)}{6}h^3 + \frac{f^{(4)}(\xi_2)}{24}h^4$$

Build an approximation of $f''(x_j)$ from a linear combination of the function values.

$$f''(x_j) \approx \alpha f(x_{j-1}) + \beta f(x_j) + \gamma f(x_{j+1}),$$

with α, β, and γ to be determined. Put the Taylor series expansions into the linear combination and collect terms

$$f(x_j)(\alpha+\beta+\gamma) + f'(x_j)(\gamma-\alpha)h + f''(x_j)(\gamma+\alpha)h^2/2 + f'''(x_j)(\gamma-\alpha)h^3/6.$$

$\alpha+\beta+\gamma=0.$

$\gamma-\alpha=0.$

$\gamma+\alpha=2/h^2.$

Solvable system

$\alpha=\frac{1}{h^2}, \beta=-\frac{2}{h^2}, \gamma=\frac{1}{h^2}.$ ✓

zero by values of $\alpha=\gamma$.

$$+ \frac{\alpha}{24}f^{(4)}(\xi_1)h^4 + \frac{\gamma}{24}f^{(4)}(\xi_2)h^4.$$

$\alpha=\gamma=1/h^2$ by IVT $= \frac{h^2}{12}f^{(4)}(\xi).$

so $$f''(x_j) = \frac{f(x_{j-1})-2f(x_j)+f(x_{j+1})}{h^2} + \frac{h^2}{12}f^{(4)}(\xi)$$

for some $\xi \in (x_{j-1}, x_{j+1})$

Consider now a grid of values on a regular grid with spacing h. Vector $\underline{F}$ with components

$$F_j = f(x_j) \qquad x_j = a + jh.$$

Now approximate $f'(x_j)$ at each grid point using centered differencing (second order).

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_{j-1})}{2h}$$

We can write this as a discrete operation

$$D_1 \underline{F} \qquad \frac{F_{j+1} - F_{j-1}}{2h}$$

components j

Similarly, we write

$$D_2 \underline{F} \qquad \frac{F_{j+1} - 2F_j + F_{j-1}}{h^2}$$

components j

as the second order approximation of the second derivative.

Math 405, Fall 2014, Day 8

We will begin our approximation of DE problems with second order, scalar boundary value problems. Here, there is an unknown function $u(x)$ that satisfies a second order equation

$$u'' = f(u, u', x) \qquad (1).$$

at each point x in an interval $[0, 1]$. More generally,

$$g(u'', u', u, x) = 0 \qquad (2)$$

but here, think of u', u, and x being given. Then (2) is a nonlinear problem for u'' which can in principle be solved (ie. by Newton's method).

In addition, two additional boundary conditions are specified

$$\left.\begin{aligned} b_0(u(0), u'(0), u(1), u'(1)) &= 0 \\ b_1(u(0), u'(0), u(1), u'(1)) &= 0 \end{aligned}\right\} \quad (3).$$

Under certain conditions, (1) & (3) have a unique solution $u(x)$ that we want to find.

Examples of equations, with $u(x)$ being (scaled) temperature with x being (scaled) position in a metal rod.

$u'' = -f(x)$ steady state heat conduction with external heating $f(x)$ per unit length.

(★) $u'' = u - f(x)$ as above but also with local Newton cooling to ambient.

$u'' = g(u) - f(x)$ $g'(u) > 1$ Nonlinear, local cooling.

$u'' = a(x)\,u' - f(x)$ Rod is hollow with a fluid moving with velocity $a(x)$.

Examples of boundary conditions in the case of (★).

$u(0) = u_0$
$u(1) = u_1$
} given end temperatures. Dirichlet conditions.

$u'(0) = 0$
$u'(1) = 0$
} Insulated ends. Homogeneous Neumann conditions.

$$u(0) = \beta u'(0)$$
$$u(1) = -\beta u'(1)$$

} ends radiate to ambient. $\beta > 0$. Robin conditions.

$$u(0) = u(1)$$
$$u'(0) = u'(1)$$

} periodic conditions (rod is bent into a ring).

Now we will turn to a finite difference scheme for (★) with periodic conditions.

Introduction to Scientific Computation: Finite Difference Methods

Brian Wetton *

September 19, 2014

1 Motivation

Many problems in Science, Engineering, and Finance involve the solution of differential equations (DE). Often these problems cannot be solved analytically but must be approximated numerically. This approximation must be done to a certain precision that depends on the application. While the differential equations do not exactly describe the real system they model (there is a modelling error) it is important to minimize the errors from the numerical approximation to be able to confirm whether the underlying mathematical model is valid. This is shown graphically in Figure 1.

2 A First DE Problem

Problem 1 *Find $u(x)$, $x \in [0,1]$ with u and its derivatives 1-periodic that satisfies*

$$-u'' + au = f(x)$$

at every x where a is a given positive constant and $f(x)$ is a given C_2 (periodic) function.

Here, C_2 (periodic) is the set of functions on the unit interval which have continuous and 1-periodic derivatives up to second order. In general, we write C_n for the set of functions that have continuous derivatives up to order n. These functions have a norm

$$\|u\|_{C_n} = \max_{0 \le j \le n} max_x |u^{(j)}(x)|$$

where $u^{(j)}$ denotes the j'th derivative of u. We will have other norms for functions, so we will label the norm we are using unless it is completely clear.

We will use the following theorem

*wetton@math.ubc.ca

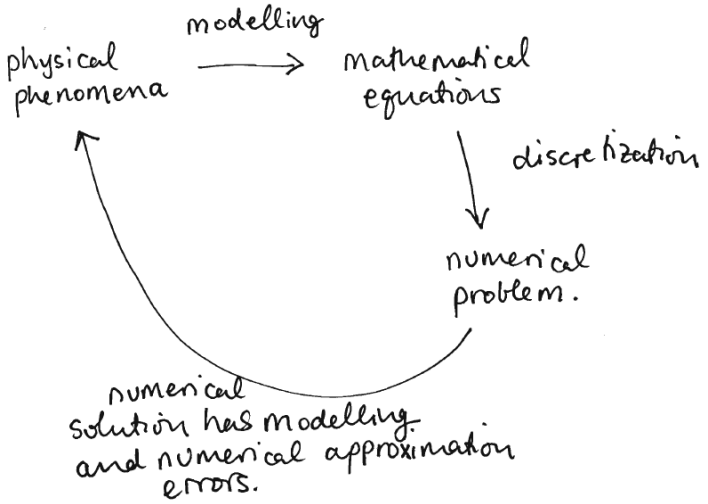


Figure 1: Stages in computational modelling. By reducing numerical errors, a clear picture of the model accuracy can be obtained.

Theorem 1 *Problem 1 has a ! (unique) solution with $u \in C_4$ with the bound*

$$\|u\|_{C_4} \leq K\|f\|_{C_2}$$

for all given $f \in C_2$ for some K that depends on a but not f.

The inequality in the Theorem is known as an a-priori bound. For a given f we don't know the solution u but we know there will be a solution, there will only be one solution, and it will have four continuous derivatives with size limited by the derivatives up to second order of f.

3 Finite Difference Discretization

3.1 Discretization

Determining the solution $u(x)$ of Problem 1 requires finding an infinite number of unknowns (the values of u at every point x in an interval). To proceed computationally, we need to deal with only a finite number of unknowns (discretization). Let's look first at a simple, Finite Difference (FD) discretization. Let U_i, $i = 1, 2, \ldots N$ approximate $u(x)$ at the ends of subintervals with length $h = 1/N$. That is

$$U_i \approx u(ih).$$

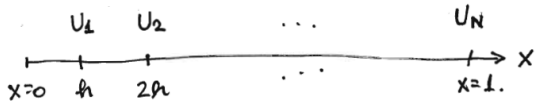


Figure 2: Uniform grid in spatial discretization.

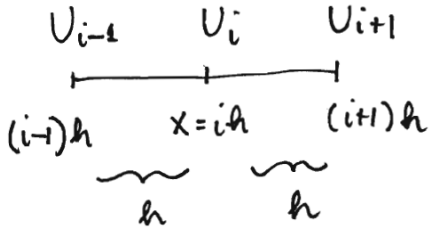


Figure 3: Values used in the finite difference approximation of the second derivative.

This is shown in Figure 2. Using the periodicity of the problem, we would have

$$U_0 = U_N \text{ and } U_{N+1} = U_1 \tag{1}$$

Convention: In these notes, I will always use lower case letters for exact solutions and upper case letters for numerically computed, approximate values.

3.2 Approximating Derivatives

Let us assume for the moment that we knew the exact solution, at least at grid points. We can use the values u_{i-1}, u_i and u_{i+1} to approximate $u''(ih)$ with the following formula:

$$D_2 u_i := \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = u''(ih) + \frac{h^2}{12} u''''(\theta) \tag{2}$$

for some $\theta \in [(i-1)h, (i+1)h]$. The geometry of this linear combination of values is shown in Figure 3. The geometry and the weights of the linear combination is called the *stencil* of the discrete approximation.

This can be derived in a number of ways, starting with Taylor Series. Two approaches are described in section 3.5 below. The last term on the right is an error term (we wanted $u''(ih)$ but got that extra term as well). Since u'''' is bounded, we can guarantee answers as accurate as we want by taking $h \to 0$

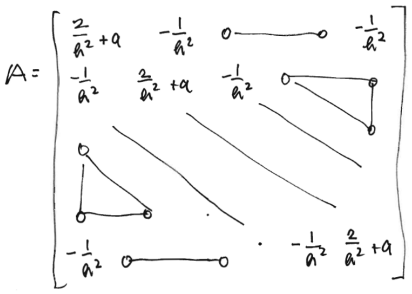


Figure 4: Structure of the matrix $\mathcal{A}$.

(that is, by refining the grid). The term h^2 in the error makes this a *second order approximation.*

3.3 Discrete Equations

Using the equation that u solves in Problem 1 we can write

$$-D_2 u_i + a u_i = f(ih) + \frac{h^2}{12} u''''(\theta) \tag{3}$$

The last term above is again an error term. We call it the *truncation error*, the residual when we put the exact solution into a discrete equation. The truncation error goes to zero as $h \to 0$. We say that the discrete equation is *consistent.* By ignoring a small (as $h \to 0$) residual, we could specify our discrete scheme for the approximate values U_i:

$$-D_2 U_i + a U_i = F_i \quad \text{for } i = 1, 2, \ldots N \tag{4}$$

where $F_i = f(ih)$ are given values. Note that (4) is a linear system with N equations for N unknowns. The system can be written in vector form

$$\mathcal{A}\mathbf{U} = \mathbf{F} \tag{5}$$

where $\mathcal{A}$ is an $N \times N$ matrix with form shown in Figure 4. We will show below that $\mathcal{A}$ is invertible, so the discrete scheme (4) has a solution $\mathbf{U}$ for every given right hand side $\mathbf{F}$ for any h.

The matrix $\mathcal{A}$ has mostly zeros. As $N \to \infty$ ($h \to 0$) the fraction of non-zeros decreases. We call such matrices *sparse.* However, $\mathcal{A}^{-1}$ is not sparse. It has no zero entries. This implies that all entries of $\mathbf{F}$ affect solution values at all locations. This is characteristic of elliptic problems.

3.4 The Next Step: Test on a Known Problem

We have a scheme for our problem. The very next thing we should do is test out the scheme on as simple a problem as we can that has a known solution. One trick is to *pick* the exact solution u, put it into the DE, and whatever residual there is call it $f(x)$. We can then use the values of f on the grid ($\mathbf{F}$) in the discrete scheme and compute $\mathbf{U}$. We can then compare $\mathbf{U}$ to the exact u at grid points to see how accurate the scheme is at various grid resolutions. For example, we could choose

$$u(x) = e^{\cos x}$$

and find

$$f(x) = (a - \sin^2 x + \cos x)e^{\cos x}$$

where we have taken the periodic interval of Problem 1 to be $[0, 2\pi]$ instead of $[0, 1]$ to make these expressions a little simpler. Choosing good test examples is somewhat of an art. You want them to represent the type of problem you are interested in, but much easier to compute (possibly lower dimensional) with an exact solution you know. In some cases, it is not possible to find an exact solution to a representative problem. In well established fields, there are often benchmark problems with high accuracy solutions you can use to test new schemes. As you are picking test problems, make sure that the solutions you use do not have zero values for the derivatives that compose the truncation error. The scheme will behave anomalously (better than usual) on such problems.

In your test, you should see how errors behave as the grid is refined (by factors of 2 for example, $h = 1/10, 1/20, 1/40, \ldots$). There are three things to look for:

1. As $h \to 0$, do the computed values U tend to the exact u values? That is, do the errors tend to zero? If so, we say that the scheme converges. The computational test is not a proof of convergence, but is strong evidence for it.

2. Is there any odd behaviour in the errors $U - u$? Odd behaviour could be an indication of a program error or it might be just a characteristic of the scheme. Odd error behaviours can be called *numerical artifacts*.

3. If the test case is (roughly) similar to problems you are actually interested in, can you achieve the desired accuracy with an N (h) value that leads to a computation that takes an acceptable length of time? If not, you should spend some time exploring ways to make the computation more efficient or more powerful computer architectures.

Testing the method on the test example above with exact solution with the parameter $a = 1$ gives the results shown in Table 1. MATLAB code for this computation is provided. Plots of the errors as functions of x for $N = 40$ and $N = 80$ are shown in Figure 5. Note that the error in the computed solution goes down by a factor of approximately 4 when N is doubled (h is halved). This matches with our discussion above, where truncation error goes down by a

N	$E_N = \max_i \lvert U_i - u(ih)\rvert$	E_N/E_{2N}
10	6.71e-2	4.24
20	1.58e-2	4.05
40	3.90e-3	4.01
80	9.73e-3	

Table 1: Errors in the finite difference method applied to the example in Section 3.4.

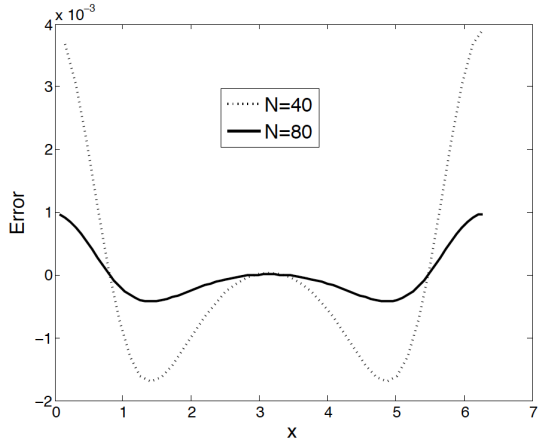


Figure 5: Spatial structure of the errors in the finite difference method applied to the example in Section 3.4.

factor of 4 when N is doubled. The exact relationship between truncation error and solution error is derived in Section 3.7 where we prove convergence of the method. Note that we have chosen to measure the error in the maximum norm

$$\|\mathbf{U}\|_\infty := \max_i |U_i|$$

in this case. Other norms could have been used and for some types of problems, other norms are more appropriate.

3.5 Derivation of FD formulae

In this section, we prove the result (2). We begin with Taylor's Polynomial Approximation Theorem:

Theorem 2 *If $f(x)$ is C_{n+1} in a neighbourhood of a then if x is in that neigh-*

bourhood,

$$f(x) = f(a)+f'(a)(x-a)+\frac{1}{2}f''(a)(x-a)^2+\cdots\frac{1}{n!}f^{(n)}(a)(x-a)^n+\frac{1}{(n+1)!}f^{(n+1)}(\theta)(x-a)^{n+1}$$

for some $\theta \in (a, x)$.

Here, the value of $f(x)$ is approximated by information at $x = a$. The first $n+1$ terms in the expression above are the n'th order Taylor Polynomial approximation of f based at $x = a$ and the last term is a remainder term, the error in the approximation. In case you have never seen the proof of this theorem, I will show the $n = 1$ case (linear approximation) in Section 3.5.1 below. We can use the Theorem to verify the properties of D_2:

$$\begin{aligned}
D_2u_i &= \frac{u_{i-1}-2u_i+u_{i+1}}{h^2} \quad (\text{let } a = ih) \\
&= \frac{1}{h^2}\left(u(a-h)-2u(a)+u(a-h)\right) \quad (\text{use the Theorem}) \\
&= \frac{1}{h^2}\Bigg(u(a)-hu'(a)+\frac{h^2}{2}u''(a)-\frac{h^3}{6}u'''(a)+\frac{h^4}{24}u''''(\theta_1) \\
&\quad -2u(a)+u(a)+hu'(a)+\frac{h^2}{2}u''(a)+\frac{h^3}{6}u'''(a)+\frac{h^4}{24}u''''(\theta_2)\Bigg)
\end{aligned}$$

where $\theta_1 \in ((i-1)h, ih)$ and $\theta_2 \in (ih, (i+1)h)$. Continuing with the expression above, we have the desired expression

$$\begin{aligned}
D_2u_i &= u''(a)+\frac{h^2}{12}\left(u''''(\theta_1)+u''''(\theta_2)\right)/2 \\
&= u''(a)+\frac{h^2}{12}u''''(\theta)
\end{aligned}$$

for some $\theta \in (\theta_1, \theta_2) \subset ((i-1)h, (i+1)h)$ where in this last step, we have used the intermediate value theorem.

This shows that the D_2 stencil has the desired properties. We could have found the coefficient values in the stencil starting with the same machinery. Taking the Ansatz

$$D_2u_i = \alpha u_{i-1}+\beta u_i+\gamma u_{i+1}$$

and wanting D_2u_i to be $u''(ih)$ with a small truncation error leads to the following requirements for the coefficients when u_{i-1} and u_{i+1} are expanded in Taylor polynomials as above:

$$\begin{aligned}
O(1) \text{ terms:} &\quad \alpha+\beta+\gamma = 0 \\
O(h) \text{ terms:} &\quad -h\alpha+h\gamma = 0 \\
O(h^2) \text{ terms:} &\quad h^2\alpha/2+h^2\gamma/2 = 1.
\end{aligned}$$

This is a linear system that can be solved for $\alpha = 1/h^2$, $\beta = -2/h^2$, $\gamma = 1/h^2$. Note that the $O(h^3)$ term also cancels with these parameters. It is typical for centred difference approximations of even derivatives that they have order of accuracy one order higher than "expected".

3.5.1 Proof of Theorem 2, $n = 1$ case

We will use Rollé's Theorem from first year calculus:

Theorem 3 *If f is differentiable in (a, b) and continuous in $[a, b]$ and $f(a) = 0$ and $f(b) = 0$ then*

$$f'(\theta) = 0$$

for some $\theta \in (a, b)$.

Consider the $n = 1$ (linear) Taylor Approximation of Theorem 2 at a specific point $x = b$. Then consider the function

$$q(x) = f(x) - L(x) - \frac{f(b) - L(b)}{(b-a)^2}(x-a)^2$$

where $L(x) = f(a) + f'(a)(x - a)$ is the linear approximation. We want to investigate $f(b) - L(b)$, the error of the linear approximation at $x = b$. This is a constant that appears in the $q(x)$ function above. Note that $q(a) = 0$ and $q(b) = 0$ so using Rollé we know that $q'(\theta_1) = 0$ for some $\theta_1 \in (a, b)$. Also, $q'(a) = 0$ so using Rollé again we have $q''(\theta) = 0$ for $\theta \in (a, \theta_1) \subset (a, b)$. We can compute

$$q''(x) = f''(x) - 2\frac{f(b) - f(a)}{(b-a)^2}$$

so $q''(\theta) = 0$ gives

$$f(b) - L(b) = \frac{f''(\theta)}{2}(b-a)^2,$$

the desired result.

3.6 Direct Solution of Sparse Linear Systems

Consider the structure of the nonzero entries of the matrix $\mathcal{A}$ in the discrete problem (5) shown in figure 4. A matrix such that

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|$$

for every row i is said to be strictly diagonally dominant. Our matrix $\mathcal{A}$ has this property. It can be shown that Gaussian elimination can be applied to such matrices stably (that is, without significant growth of floating point round-off errors) without pivoting. It can be seen that Gaussian elimination and back substitution can be done for the system (5) with a finite number of operations per row independent of the number of rows. The structure of the LU decomposition of $\mathcal{A}$ is shown in figure 6. The total operation count to find the solution is $O(N)$, that is the operations are bounded by a constant times N. Thus a direct solver applied to this problem taking into account the sparsity of $\mathcal{A}$ has optional complexity.

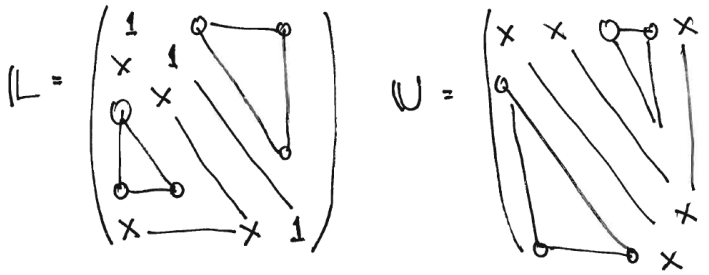


Figure 6: Structure of the LU decomposition of the matrix A.

Note: This is close to the ideal example for sparse numerical linear algebra. The matrix A is (almost) narrowly banded, is diagonally dominant, is symmetric and positive definite. We will see that the situation for discretizations in higher dimensional problems is not as ideal. Still, it is possible now to solve 3D problems of *modest* size with direct solvers on basic computers.

3.7 Convergence Proof

We came up with our discrete scheme (4) by neglecting a small truncation error in relationships the exact solution at the grid points satisfy (3):

$$\begin{aligned} \mathcal{A}\mathbf{U} &= \mathbf{F} \\ \mathcal{A}\mathbf{u} &= \mathbf{F} + \tau \end{aligned}$$

where $\tau_i = h^2 u^{(4)}(\theta_i)/12$ are the truncation errors at every grid point. As discussed above, the truncation errors go to zero as $h \to 0$ (the definition of a consistent scheme). The vector of errors in the computed solutions at grid points is

$$\mathbf{E} = \mathbf{U} - \mathbf{u}.$$

For this linear problem, we can take the difference of the two equations above to obtain

$$\mathcal{A}\mathbf{E} = \tau \quad \text{or} \quad \mathbf{E} = \mathcal{A}^{-1}\tau. \tag{6}$$

Note: We haven't shown yet that A is invertible (but our numerical implementation suggests it is for all h) and in practice we would never compute the full matrix $\mathcal{A}^{-1}$. This is just a representation for the theory.

Considering (6) we see that with τ small, $\mathbf{E}$ will be small as long as multiplying by $\mathcal{A}^{-1}$ does not increase its size by more than a constant (independent of h). Formally, we want to have the following property:

$$\|\mathcal{A}^{-1}\mathbf{b}\|_\infty \leq C\|\mathbf{b}\|_\infty \tag{7}$$

for all $\mathbf{b}$ with C independent of h. Here,

$$\|\mathbf{b}\|_\infty := \max_{i=1,\ldots N} |b_i|$$

is the maximum norm of vectors. For fixed h we can define

$$\|\mathcal{A}^{-1}\|_\infty := \max_{\mathbf{b}\neq\mathbf{0}} \frac{\|\mathcal{A}^{-1}\mathbf{b}\|_\infty}{\|\mathbf{b}\|_\infty}.$$

This is known as an induced matrix norm. The value $\|\mathcal{A}^{-1}\|_\infty$ is the smallest value C for fixed N such that (7) holds for all $\mathbf{b}$. The property we are looking for is then that

$$\sup_h \|\mathcal{A}^{-1}\|_\infty$$

is finite. This property defines *maximum norm stability* of the scheme. Showing stability of numerical schemes in general is quite difficult, but easy for this particular scheme. Consider the left hand equation of (6) where we look back to (4) to see the details of the matrix $\mathcal{A}$:

$$\begin{aligned} -D_2E_i + aE_i &= \tau_i \\ -\frac{1}{h^2}E_{i-1} + (\frac{2}{h^2} + a)E_i - \frac{1}{h^2}E_{i+1} &= \tau_i. \end{aligned} \tag{8}$$

Suppose that the $\max_i |E_i|$ is attained at an index j and that $E_j > 0$. Thus, $E_{j-1} \leq E_j$ and $E_{j+1} \leq E_j$ and thus

$$2E_j - E_{j+1} - E_{j-1} \geq 0$$

When this used in (8) we find that

$$aE_j \leq \tau_j \quad \text{or} \quad E_j \leq \frac{1}{a}|\tau_j|.$$

Since $\|\mathbf{E}\|_\infty = E_j$ we have

$$\|\mathbf{E}\|_\infty \leq \frac{1}{a}\|\tau\|_\infty.$$

Since $\mathbf{E} = \mathcal{A}^{-1}\tau$ we have shown the maximum norm stability of the scheme. If $\|\mathbf{E}\|_\infty$ is attained at an index j where $E_j < 0$, a similar argument applies.

Recall that $\tau_i = \frac{h^2}{12}u^{(4)}(\theta_i)$ for some $\theta_i \in ((i-1)h, (i+1)h)$ so

$$\|\tau\|_\infty \leq \frac{\|u\|_{C_4}}{12}h^2.$$

using Theorem 1 we have

$$\|\tau\|_\infty \leq \frac{K\|f\|_{C_2}}{12}h^2.$$

Using the stability result we derived,

$$\|\mathbf{E}\|_\infty \leq \frac{K\|f\|_{C_2}}{12a}h^2. \tag{9}$$

This proves the convergence of the scheme since as $h \to 0$, $\|\mathbf{E}\|_\infty \to 0$. With $\|\mathbf{E}\|_\infty < Ch^2$ we say that the convergence is *second order*.

The analysis leading to the convergence result above is an example of the Lax Equivalence Theorem for linear problems, often stated informally as

```
Consistency + Stability = Convergence
```

Note that in (9) that if $\|f\|_{C_2}$ is large (*i.e.* f is highly oscillatory) then h must be quite small to make the errors small. This makes sense: to resolve oscillatory behaviour, a fine grid is needed.

3.8 von Neumann Analysis

Consider again our discretization of Problem 1,

$$\mathcal{A}\mathbf{U} = \mathbf{F}, \quad \mathcal{A} = -D_2 + aI.$$

It is possible to show the stability of the scheme in other norms. Here, we will consider the l_2 norm, also known as the energy norm or the (scaled) Euclidean norm

$$\|\mathbf{b}\|_2 := \sqrt{h \sum_{i=1}^{N} |b_i|^2}. \tag{10}$$

Note that the scaling by h is such that if $b_i = C$ for all i, then $\|\mathbf{b}\|_2 = C$ for every N (h) since $N = 1/h$. Also, this makes the norm analogous to the continuous energy norm for the space of functions L_2:

$$\|f\|_{L_2} = \sqrt{\int_0^1 |f(x)|^2 dx}.$$

The discrete l_2 norm (10) can be seen as an approximation (trapezoidal rule) of the continuum norm.

3.8.1 Discrete Fourier vectors

To proceed, we introduce the complex valued Discrete Fourier (DF) vectors $\mathbf{f}_\alpha$:

$$f_{\alpha,j} = e^{2\pi i \alpha j h}.$$

To clarify the labelling, $\mathbf{f}_\alpha$ is a vector with N complex components for every $\alpha = 0, \ldots N-1$. The N components are indexed by j above and the i is the pure imaginary unit. The set $\{\mathbf{f}_\alpha\}$ is a basis of $\mathbf{C}_N$, orthonormal in the inner product that corresponds to the l_2 norm:

$$\begin{aligned} (\mathbf{a}, \mathbf{b}) &:= h \sum_{j=1}^{N} a_j b_j^* \\ \text{so } (\mathbf{a}, \mathbf{a}) &= \|\mathbf{a}\|_2^2 \end{aligned}$$

Since the DF vectors are a basis, we can write any vector as a linear combination of these vectors, that is

$$\mathbf{a} = \hat{a}_0\mathbf{f}_0 + \hat{a}_1\mathbf{f}_1 + \ldots + \hat{a}_{N-1}\mathbf{f}_{N-1}$$

for ! coefficients $\hat{\mathbf{a}}$, the scaled DF transform of $\mathbf{a}$. We are pursuing some theoretical properties here, but there are practical situations where $\hat{\mathbf{a}}$ is desired and it can be computed efficiently using the Fast Fourier Transform algorithm. Since $\{\mathbf{f}_\alpha\}$ is an orthonormal basis,

$$\|\mathbf{a}\|_2 = \|\hat{\mathbf{a}}\|_2. \tag{11}$$

3.8.2 Application to discretizations

It can be shown that the DF vectors are always the complete set of eigenvectors of any linear, constant coefficient, periodic, finite difference discretization on a uniform grid. As an example, this will be shown explicitly for our discretization of Problem 1.

$$\begin{aligned} D_2 f_{\alpha,j} &= \frac{1}{h^2}(f_{\alpha,j-1} - 2f_{\alpha,j} + f_{\alpha,j+1}) \\ &= \frac{1}{h^2}e^{2\pi i\alpha jh}\left(e^{-2\pi i\alpha h} - 2 + e^{2\pi i\alpha h}\right) \\ &= \frac{2}{h^2}(\cos(2\pi\alpha h) - 1)f_{\alpha,j}. \end{aligned}$$

Thus

$$D_2\mathbf{f}_\alpha = \frac{2}{h^2}(\cos(2\pi\alpha h) - 1)\mathbf{f}_\alpha$$

and

$$\mathcal{A}\mathbf{f}_\alpha = \lambda_\alpha\mathbf{f}_\alpha$$

with $\lambda_\alpha = \frac{2}{h^2}(1 - \cos(2\pi\alpha h)) + a$. The set of the eigenvalues $\{\lambda_\alpha\}$ of $\mathcal{A}$ corresponding to the DF vectors is called the *symbol* of $\mathcal{A}$. Consider again our discretization:

$$\mathcal{A}\mathbf{U} = \mathbf{F}.$$

We can write it in terms of the the DF components

$$\hat{U}_\alpha = \frac{1}{\lambda_\alpha}\hat{F}_\alpha$$

where we have gained considerable insight from diagonalizing the problem. The eigenvalues λ_α are all positive and $\geq a$ so

$$\begin{aligned} & |\hat{U}_\alpha| \leq \frac{1}{a}|\hat{F}_\alpha| \quad \text{for every } \alpha \\ \Rightarrow \quad & \|\hat{\mathbf{U}}\|_2 \leq \frac{1}{a}\|\hat{\mathbf{F}}\|_2 \\ \Rightarrow \quad & \|\mathbf{U}\|_{\mathbf{2}} \leq \frac{\mathbf{1}}{\mathbf{a}}\|\mathbf{F}\|_{\mathbf{2}} \end{aligned}$$

using (11). This shows the l_2 norm stability of the scheme. Applying the same result to $\mathcal{A}\mathbf{E} = \tau$ shows the l_2 convergence of the scheme,

$$\|\mathbf{E}\|_2 \leq Ch^2$$

using the bounds on the truncation error τ from the previous section. Note that a (sub-optimal) maximum norm convergence result can be derived from this, since

$$\begin{aligned} & h|E_i|^2 \leq h\sum_{j=1}^{N}|E_j|^2 := \|\mathbf{E}\|_2^2 \leq (Ch^2)^2 \text{ for each } i \\ \Rightarrow \quad & |E_i|^2 \leq C^2h^3 \text{ for each } i \\ \Rightarrow \quad & \|\mathbf{E}\|_\infty \leq Ch^{3/2} \end{aligned}$$

So from von Neumann analysis we can show that scheme does converge in maximum norm, but at the non-optimal rate of 3/2. We know from our numerical test that second order accuracy is maximum norm is observed and confirmed that in our first, maximum norm, stability analysis.

Remark: Sometimes there is a gap between what can be proved and the actual behaviour of the method.

3.9 Implementing Boundary Conditions

Consider Problem 1 in the interval [0,1], but with local boundary conditions specified at the ends $x = 0$ and $x = 1$ rather than periodic conditions. Possible conditions for this problem at $x = 0$ are

$$u(0) = a, \quad a \text{ given (Dirichlet)} \tag{12}$$

$$u'(0) = a \quad \text{(Neumann)} \tag{13}$$

$$u'(0) - \alpha u(0) = a, \quad \alpha > 0 \text{ given (Robin).} \tag{14}$$

Similar conditions can be given at $x = 1$. For (14) at $x = 1$, $\alpha < 0$ for physically stable models. In this setting with a uniform grid with spacing $h = 1/N$ we will in general have $N + 1$ discrete unknowns $U_0, U_1, \ldots U_N$ at $x = 0, h, \ldots 1$. If we have Dirichlet conditions at both interval ends, we can apply $U_0 = a$ directly, and the same for U_N. The value of U_0 only appears in the stencil at grid point 1:

$$D_2U_1 = \frac{U_0 - 2U_1 + U_2}{h^2} = \frac{-2U_1 + U_2}{h^2} + \frac{a}{h^2}.$$

In the implementation, the first two terms of the right expression above become part of the matrix $\mathcal{A}$ and the last term contributes to the right hand side vector. In this case, the end point values have been eliminated and the system to be solved is of size $N - 1$.

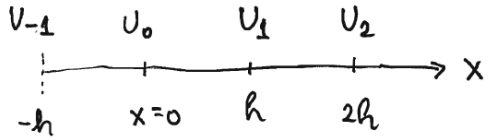


Figure 7: Grid points near the $x = 0$ boundary.

3.9.1 First derivative approximations

To proceed to the other conditions (Neumann and Robin) we need to discuss finite difference approximations of the first derivative.

$$
\begin{aligned}
D_+U_j &:= \frac{U_{j+1} - U_j}{h} = u'(jh) + \frac{h}{2}u''(jh) + \dots \text{ (forward differencing, first order)} \\
D_-U_j &:= \frac{U_j - U_{j-1}}{h} = u'(jh) - \frac{h}{2}u''(jh) + \dots \text{ (backward differencing, first order)} \\
D_1U_j &:= \frac{U_{j+1} - U_{j-1}}{2h} = u'(jh) + \frac{h^2}{6}u'''(jh) + \dots \text{ (centred differencing, second order)} \\
\tilde{D}_+U_j &:= \frac{-\frac{3}{2}U_j + 2U_{j+1} - \frac{1}{2}U_{j+2}}{h} = u'(jh) + \frac{h^2}{3}u'''(jh) + \dots \text{ (second order forward differencing)}
\end{aligned}
$$

Notice that although both D_1 and $\tilde{D}_+$ are second order accurate, $\tilde{D}_+$ has a larger error constant. Note also that $D_1 = \frac{1}{2}(D_+ + D_-)$ and that this combination cancels the first order error terms.

3.9.2 Implementing Neumann conditions

Consider now implementing the Neumann condition (13). There are several approaches. Since implementing boundary conditions is often a source of confusion, I will go through some of the options in detail. In what follows, refer to Figure 7 for the numbering of the unknowns.

U_0 **equation for Neumann condition:** In this scenario, U_0 remains an unknown and we use an approximation of the Neumann condition for its corresponding discrete equation. Using first order differencing

$$D_+U_0 := \frac{U_1 - U_0}{h} = a$$

leads to approximate values that are only first order accurate at *all* grid points. We can easily maintain global second order accuracy by using

$$\tilde{D}_+U_0 = a \tag{15}$$

instead.

U_0 eliminated using one sided differencing: We notice again that U_0 only appears in the U_1 equation and so we can use (15) to eliminate it:

$$\begin{aligned} D_2U_1 &:= \frac{U_0 - 2U_1 + U_2}{h^2} \\ &= \frac{\frac{2}{3}(2U_1 - \frac{1}{2}U_2 - ha) - 2U_1 + U_2}{h^2} \\ &= \frac{2(U_2 - U_1)}{3h^2} - \frac{2a}{3h} \end{aligned} \tag{16}$$

However, (16) can cause some confusion since the truncation error in (15) is second order but (16) is only first order accurate. Analytically, (16) is the right way to view the system since the corresponding $(N+1)\times(N+1)$ matrices are max-norm stable.

Introduce the ghost point U_{-1}: You can also introduce the ghost point U_{-1} as shown in Figure 7. Consider U_{-1} to approximate the solution extended from the interior to $x = -h$ using a Taylor polynomial of high enough order. The Neumann condition $u'(0) = a$ can then be approximated by

$$D_1U := \frac{U_1 - U_{-1}}{2h} = a.$$

This equation and U_{-1} can be added to the system or U_{-1} can be eliminated using this condition as done above. This approximation has a smaller truncation error constant than the second order one-sided approach and so is preferred.

Robin conditions can be implemented in a similar manner.

3.9.3 Implementing boundary conditions for staggered grid discretizations

We can consider approximate values at subinterval centres rather than subinterval ends. For second order methods, this is equivalent to considering unknowns that are the integral average of the unknown function over the subinterval (the basis of finite volume methods). Values on this grid near the $x = 0$ boundary are shown in Figure 8. Here, a ghost value is needed even for a Dirichlet condition, which is then approximated to second order using linear interpolation (averaging):

$$\frac{U_{1/2} + U_{-1/2}}{2} = a \quad \text{approximates } u(0) = a$$

and short centred differencing is used for a Neumann condition

$$\frac{U_{1/2} - U_{-1/2}}{h} = a \quad \text{approximates } u'(0) = a.$$

Note that this approximation has a dominant truncation error term of $\frac{h^2}{24}u'''(0)$. This is the most accurate second order way to approximate Neumann conditions

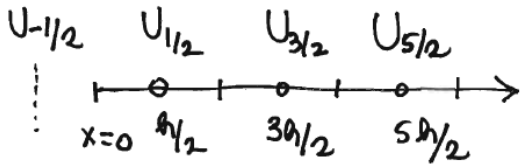


Figure 8: Staggered grid points near the $x = 0$ boundary.

in this finite difference framework. As discussed above, these equations can be incorporated into a matrix of the discretization $\mathcal{A}$, or the ghost value $U_{-1/2}$ can be eliminated.

3.10 Asymptotic Error Analysis

Our convergence proof for the discretization of the periodic problem showed that

$$\|\mathbf{E}\|_\infty \leq Ch^2$$

where $\mathbf{E} = \mathbf{U} - \mathbf{u}$. Computationally, we saw that in fact

$$E_i = c(ih)h^2 + o(h^2)$$

with an underlying smooth function $c(x)$ independent of h. Above, $o(h^2)$ (notice the lower case o) is a quantity smaller than any constant times h^2 as $h \to 0$. For smooth data f this result is not hard to show and the remainder term $o(h^2)$ is $O(h^4)$. Let us see what $c(x)$ would have to be to have the property

$$U_i = u(ih) + c(ih)h^2 + \ldots \tag{17}$$

This is known as an asymptotic error expansion. Plug this into the discrete equations $-D_2U_i + aU_i = F_i$ and use the remainder terms for the approximation for D_2 we derived in Section 3.5:

$$-u'' - \frac{h^2}{12}u'''' + O(h^4) - h^2c'' + au + ah^2c = f \tag{18}$$

where these terms are all evaluated at $x = ih$. If (18) is to hold for all h, then the coefficients of powers of h must match:

$$O(1): \qquad -u'' + au = f \quad \text{with } u \text{ periodic} \tag{19}$$

$$O(h^2): \qquad -c'' + cu = \frac{1}{12}u'''' \quad \text{with } c \text{ periodic.} \tag{20}$$

Equation (19) is satisfied by the exact solution. We don't have to actually solve (20) but we do know that this problem has a smooth solution $c(x)$. Note that

$c(x)h^2$ is the dominant error term (17) and from (20) we see that $c(x)$ is the response of the system to a RHS that is the truncation error. This makes sense.

All of this may seem like formal so far, but now consider

$$\tilde{\mathbf{E}} = \mathbf{U} - (\mathbf{u} + h^2\mathbf{c}).$$

We have

$$\mathcal{A}\tilde{\mathbf{E}} = O(h^4)$$

using (19) and (20). Using the stability result for $\mathcal{A}$ from Section 3.7 we have

$$\tilde{\mathbf{E}} = O(h^4).$$

This shows that (17) is accurate to fourth order, the desired result.

There are several consequences of the result

$$\mathbf{U} = \mathbf{u} + h^2\mathbf{c} + O(h^4)$$

with $c(x)$ smooth, independent of h:

Richardson Extrapolation: This justifies Richardson extrapolation

$$\tilde{\mathbf{U}}_h := \frac{4}{3}\mathbf{U}_{h/2} - \frac{1}{3}\mathbf{U}_h = \mathbf{u} + O(h^4)$$

where in the expression above $\mathbf{U}_h$ is a coarse grid computation and $\mathbf{U}_{h/2}$ are values from a fine grid (refined by a factor of 2) computation taken at coarse grid points. Note that this is not an efficient way to make a fourth order method. In addition, it is not reliable since not all schemes have regular errors like this one.

Discrete Smoothness: This result also shows that derivative approximations converge with full order. Consider our basic estimate

$$\mathbf{U} = \mathbf{u} + O(h^2).$$

If we were interested in values of the first derivative of the solution, we would compute

$$D_1\mathbf{U} = D_1\mathbf{u} + O(h) = \mathbf{u}' + O(h)$$

where the order of h is lost because we divide the $O(h^2)$ by h when we apply D_1 and we have not taken into account the structure of the $O(h^2)$ term. However, if we know the asymptotic error result applies

$$\mathbf{U} = \mathbf{u} + h^2\mathbf{c} + O(h^4)$$

with $c(x)$ smooth then we see that

$$D_1\mathbf{U} = D_1\mathbf{u} + h^2 D_1\mathbf{c} + O(h^3) = \mathbf{u}' + h^2(\mathbf{c}' + \frac{1}{6}\mathbf{u}''') + O(h^3) = \mathbf{u}' + O(h^2)$$

and we see convergence order preserved for derivatives. In Finite Element Method (FEM) literature, this "unexpected" increase in convergence order is sometimes called "superconvergence" (although this term also has other meanings).

Discrete Embedding: An asymptotic error expansion can also overcome deficiencies in the stability analysis using weak norms. Consider the conversion of the l_2 estimate to the maximum norm estimate considered in Section 3.8. Following that argument with

$$\tilde{\mathbf{E}} := \mathbf{U} - (\mathbf{u} + h^2\mathbf{c}) = O(h^4)$$

gives

$$\begin{aligned} \|\tilde{\mathbf{E}}\|_\infty &\leq Ch^{7/2} \\ \|\mathbf{E} - h^2\mathbf{c}\|_\infty &\leq Ch^{7/2} \\ \|\mathbf{E}\|_\infty &= O(h^2) \end{aligned}$$

where in the last step we have used the triangle inequality.

The existence of regular asymptotic error behaviour (and so all the results above) relies on the problem having smooth solutions and being computed on a regular (structured) grid. This shows one of the advantages of using structured meshes. In addition, discretizations on structured meshes are more efficiently implemented, especially on specific computational architectures like Graphical Processing Units (GPUs). However, structured meshes restrict local adaptivity and do not apply in a straightforward way to general problem geometries in higher dimensions.

Math 405/607, Fall 2014, Day 10

approximations to boundary value problems can also be done with Finite Volume and Finite Element methods. FEM are the subject of Math 521 next term.

For FVM, the discrete unknowns are not approximate values at grid points, but approximations to integrals of the solution over subintervals.

$$U_{j-1/2} \approx \int_{x_{j-1}}^{x_j} u(x)\,dx \approx h\, u(x_{j-1/2}) \qquad j = 1, \cdots N$$

↑ $x_j = a + jh$. $h = (b-a)/N$.

→ Midpoint rule approximation.

Consider

$$-(a(x)u')' + u = f(x) \qquad a(x) > 0, \text{ given for all } x$$

↑ variable conductivity of the rod material.

Integrate over $[x_{j-1}, x_j]$.

$$-a(x)u'\Big|_{x_j}^{x_{j+1}} + U_{j-1/2} \approx \int_{x_j}^{x_{j+1}} f(x)\,dx \approx h\, f(x_{j-1/2}) \qquad (1)$$

→ midpoint rule approximation.

Now $u'\big|_{x_j} \approx \dfrac{U_{j-1/2} - U_{j-3/2}}{h}$ (short centered differencing).

So (1) becomes the discrete scheme

$$-a(x_j)\frac{U_{j+1/2} - U_{j-1/2}}{h} + a(x_{j-1})\frac{U_{j-1/2} - U_{j-3/2}}{h} + U_{j-1/2} = h\,F(x_{j-1/2}). \qquad (2)$$

Note that if $a(x) \equiv 1$ and we divide the above equation by h, we have the same discrete equations as we had for the FD approximation.

Note that the linear system for the discretization is symmetric (desireable). If we began with

$$-(a(x)\,u')' + u = F(x) \qquad (\star).$$

$$\Rightarrow \quad -a(x)u'' - a'(x)\,u' + u = F(x)$$

and discretized with standard FD methods,

$$-a_j D_2 U_j - a'_j D_1 U_j + U_j = F_j$$

the resulting system is not symmetric. The form $(\star)$ of the equations is known as divergence - or conservative-form.

Consider now a general scalar boundary value problem:

$$u'' = g(x, u, u') \quad \leftarrow \text{consider arguments } g(x, u, v) \text{ in partial derivatives below.}$$

This can be discretized in a natural way using FD approximations:

$$D_2 U_j = \frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} \qquad D_1 U_j = \frac{U_{j+1} - U_{j-1}}{2h}$$

$$D_2 U_j = g(x_j, U_j, D_1 U_j).$$

This is a nonlinear system for the unknowns $\underline{U}$, which can be written

$$\underline{N}(\underline{U}) = \underline{0}, \qquad N_i = D_2 U_i - g(x_i, U_i, D_1 U_i).$$

We could use Newton's method to solve it

$$\underline{U}^{(n+1)} = \underline{U}^{(n)} - [\mathbb{J}]^{-1} \underline{N}(\underline{U}^{(n)}).$$

$\underline{U}^{(n)}$ is the n'th iterate of Newton's method

where $\mathbb{J}$ is the jacobian matrix $J_{ij} = \frac{\partial N_i}{\partial U_j}$

$$J_{ij} = \begin{cases} 0 & \text{if } j \neq i, i-1, i+1. \\ \frac{1}{h^2} - \frac{\partial g}{\partial v}(x_i, U_i, D_1 U_i)\frac{1}{2h}. & j = i+1. \\ \frac{1}{h^2} + \frac{\partial g}{\partial v}(x_i, U_i, D_1 U_i)/2h & j = i-1 \\ \frac{-2}{h^2} - \frac{\partial g}{\partial u}(x_i, U_i, D_1 U_i) & j = i. \end{cases}$$

Note that $\underline{\underline{I}}$ here has the same sparse structure here that the FD discretization matrix $\underline{\underline{A}}$ had from our discretization of the simple problem

$$-u'' + u = F(x) \quad , \text{ approximated by } \quad -D_2\underline{U} + \underline{U} = \underline{F}.$$

FD, FV and FE methods naturally extend to problems in higher dimensions. If the domains are complicated geometrically, FEM are the most natural.

In 1D, "domains" are all intervals and there are techniques specialized to this case, which we will consider here.

Consider the original, periodic problem

$$-u'' + u = F(x)$$

Take the problem to be 2π periodic for convenience of writing the Fourier series solution

$$u(x) = \sum_{\alpha=-\infty}^{\infty} \hat{u}_\alpha e^{i\alpha x}$$

where $\hat{u}_\alpha = \dfrac{\hat{f}_\alpha}{\alpha^2 + 1}$.

$$\hat{f}_\alpha = \frac{1}{2\pi} \int_0^{2\pi} F(x) e^{-i\alpha x} \, dx.$$

We can approximate this with spectral accuracy as follows:

$$\underline{U} = \mathbb{F}^{-1} \Lambda \mathbb{F} \underline{F}. \qquad (3)$$

↑ inverse DFT. ↑ diagonal matrix, values $\frac{1}{\Lambda_{\alpha\alpha}} = 1+\alpha^2$ (aliased). ↑ DFT ↑ values of $f(x_j)$ on the grid.

Note that the matrix

$$\widehat{D}_2 = \mathbb{F}^{-1} \text{diag}(-\alpha^2) \mathbb{F}$$ is a full matrix,

↑ aliased

$\widehat{D}_2 \underline{U}$ computes values of u'' on the grid to spectral accuracy. However, $\widehat{D}_2$ is never calculated explicitly in (3) in which each of the 3 matrix multiplications can be done quickly.

Non-periodic problems can be handled using Chebyshev spectral approximation.

For many problems, local adaptivity in interval size is more effective to obtaining accurate solutions than high order approximation on a uniform grid. We will discuss one such method today, the basis of MATLAB's

bvp4c routine. This approximation requires the ODE to be written as a first order system. We'll also need to do this for our next topic of numerical methods for initial value problems. DE's of order m can always be converted to first order systems with m components.

Ex $-u'' + u = F(x)$.

Introduce $V(x) = u'(x)$. The equivalent system is

$$u'(x) = V(x)$$
$$V'(x) = u(x) - F(x).$$

↑

$$u'' = u - F(x)$$

$u(x)$ solution is the same, $V(x) = u'(x)$ ✓

Ex $u'' = g(x, u, u')$.

Introduce $V(x) = u'(x)$

$u'(x) = V(x)$ (always the same)

$V' = g(x, u, V)$.

Ex $u(t)$, $\frac{d^4u}{dt^4} - 2\frac{d^3u}{dt^3} + 4\frac{du}{dt} - u = f(t)$.

Introduce $u_1(t) = \dot{u}(t)$, $u_2(t) = \ddot{u}(t)$, $u_3(t) = \dddot{u}(t)$

$\dot{u} = u_1$

$\dot{u}_1 = u_2$

$\dot{u}_2 = u_3$

$\frac{d^4u}{dt^4} \longrightarrow \dot{u}_3 = 2u_3 - 4u_1 + u + f(t)$.

So we can consider a boundary value problem written as a first order system with m components

$$\underline{S}'(x) = \underline{F}(x, \underline{S}(x)). \qquad (4).$$

↑ m components, $x \in [0,1]$

Also, m additional conditions must be given, involving $\underline{S}(0)$, $\underline{S}(1)$, $\underline{S}'(0)$, $\underline{S}'(1)$. As usual, divide the interval $[0,1]$ into N subintervals. Consider equal sized subintervals $h = 1/N$ for simplicity, but the technique described below is amenable to adaptive discretizations (h made smaller where needed for accuracy).

The $m(N+1)$ values of $\underline{S}$ at subinterval ends (grid points) are the unknowns in this discretization. For given values of $\underline{S}$ at grid points, $\underline{S}'$ at grid points is determined through (4). Thus, we can make cubic (Hermite) approximations for the components of $\underline{S}$ in subintervals.

Summary: grid point values of $\underline{S}$ give piecewise C_1 cubic polynomials $\underline{C}(x)$ in subintervals. (4) is satisfied by $\underline{C}(x)$ at subinterval ends by construction. We have mN equations resulting from satisfying (4) with $\underline{C}(x)$ at subinterval centres (this is known as a collocation method):

$$\underline{C}'(x_*) = \underline{F}(x_*, \underline{C}(x_*)) \qquad (5)$$

for each subinterval centre x_*. We obtain another m conditions from the boundary conditions. Together that gives $m(N+1)$ equations for the $m(N+1)$ unknown grid values of $\underline{S}$.

Consider the truncation error of (5), looking at one component $C(x)$, $x \in [x_{j-1}, x_j]$. $(x_j - x_{j-1}) = h$.

As usual, map $x \mapsto y \in [0,1]$,

$$y = \frac{x - x_j}{h}. \qquad \frac{dy}{dx} = \frac{1}{h}, \quad \frac{dx}{dy} = h.$$

$$\begin{aligned} C(y) = {} & S(0)(1 - 3y^2 + 2y^3) \\ & + S(1)(3y^2 - 2y^3) \\ & + S'(0)(y - 2y^2 + y^3) \\ & + S'(1)(y^3 - y^2) \end{aligned}$$

Consider $S(1/2) - C(1/2)$ by constructing

$$q(y) = S(y) - C(y) - \underbrace{16\, y^2(y-1)^2}\,\big(S(1/2) - C(1/2)\big).$$

value 1 at $y = 1/2$,
Value and derivative zero at $y = 0, 1$.

$q(y) = 0$ at $y = 0, 1/2, 1$, Ao
$q'(\xi_j) = 0 \quad \xi_1 \in (0, 1/2), \ \xi_2 \in (1/2, 1)$.

Also, $q'(0) = q'(1) = 0$, so

$q''(\xi) = 0$ at three places, one zero in each of

$(0, \xi_1)$, (ξ_1, ξ_2), $(\xi_2, 1)$

so $q''''(\xi) = 0$ at some $\xi \in (0,1)$

$$q''''(y) = S''''(y) - 16 \cdot 24 \left(S(1/2) - C(1/2) \right)$$

so $$S(1/2) - C(1/2) = \frac{d^4 S}{dy^4} \Big/ 384$$

Scaling back to the original interval,

$$S(x_*) - C(x_*) = \frac{h^4}{384} \frac{d^4 S}{dx^4}$$

So replacing $S(x_*)$ by $C(x_*)$ on the right hand side of (5) makes a fourth order truncation error. What about $S'(x_*) - C'(x_*)$?

In the reference interval, (6).

$$C'(1/2) = \frac{3}{2} \left(S(1) - S(0) \right) - \frac{1}{4} \left(S'(0) + S'(1) \right)$$

Using Taylor series about $y = 0$:

$$S'(1/2) = S'(0) + \frac{1}{2} S''(0) + \frac{1}{8} S'''(0) + \frac{1}{48} S''''(0) + \frac{1}{16 \cdot 24} S^{(5)}(\xi_1)$$

$$S(1) = S(0) + S'(0) + \frac{1}{2} S''(0) + \frac{1}{6} S'''(0) + \frac{1}{24} S^{(4)}(0) + \frac{1}{120} S^{(5)}(\xi_2)$$

$$S'(1) = S'(0) + S''(0) + \frac{1}{2}S'''(0) + \frac{1}{6}S^{(4)}(0) + \frac{1}{24}S^{(5)}(\xi_3).$$

Collect terms in (6):

$$C'(1/2) = S(0)\cdot\frac{3}{2}(1-1)^0 \quad \checkmark$$

$$+ S'(0)\left(\frac{3}{2} - \frac{1}{4} - \frac{1}{4}\right)^1 \quad \checkmark.$$

$$+ S''(0)\left(\frac{3}{4} - \frac{1}{4}\right)^{1/2} \quad \checkmark.$$

$$+ S'''(0)\left(\frac{3}{2}\left(\frac{1}{6}\right) - \frac{1}{4}\frac{1}{2}\right)^{1/8} \quad \checkmark.$$

$$+ S''''(0)\left(\frac{3}{2}\frac{1}{24} - \frac{1}{4}\frac{1}{6}\right)$$

$$\underbrace{\quad\quad}$$

$$\frac{1}{6} - \frac{1}{24} = \frac{3-2}{48} \quad \checkmark$$

Note: $y = 1/2$ is a "sweet spot" where an extra order of accuracy is obtained.

Thus $|C'(1/2) - S'(\tfrac{1}{2})| \le A \max \frac{d^5 S}{dy^5}$.

Scaling back to the original interval,

$$h\left|\frac{dC}{dx}(x_*) - \frac{dS}{dx}(x_*)\right| \le A h^5 K_5.$$

so $$\left|\frac{dC}{dx}(x_*) - \frac{dS}{dx}(x_*)\right| = O(h^4)$$

So now consider the truncation error in (5).

$$|C'(x_*) - f(x_*, C(x_*))| = \cdots$$

where as above, $C(x)$ is constructed using the exact solution at grid points.

$$\cdots = \left| C'(x_*) - f(x_*, C(x_*)) \right| - \left(S'(x_*) - f(x_*, S(x_*) \right.$$

$$\leq \left| C'(x_*) - S'(x_*) \right| + \left| f(x_*, C(x_*) - f(x_*, S(x_*)) \right|$$

$$\leq O(h^4) + \frac{\partial f}{\partial C}(x_*, \theta) \left| C(x_*) - S(x_*) \right|$$

θ: between $C(x_*)$ and $S(x_*)$

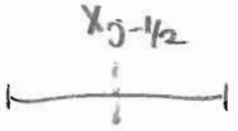
$$= O(h^4)$$

So we have shown the interior equations are consistent with $O(h^4)$ truncation error. Next time, we will evaluate the stability of the scheme applied to our original test problem on a regular grid.

Math 405/607, Fall 2014, Day 10 update

Replacing pages 1-2 from the posted notes.

Finite Volume Methods use discrete unknowns that are approximations of the integrals of solutions over subintervals:



$\frac{x_j + x_{j-1}}{2}$ ↓

$$U_{j-1/2} \approx \int_{x_{j-1}}^{x_j} u(x)\,dx \approx h\,u(x_{j-1/2})$$

midpoint rule

↑ makes the grid index $j-1/2$ appropriate.

Consider $-(a(x)\,u')' + u = F(x)$ (1)

$a(x)$ given, $a(x) > 0$ for all x

Integrate over $[x_{j-1}, x_j]$

$$-a(x)\,u'\Big|_{x_j}^{x_{j+1}} + U_{j-1/2} \approx \int_{x_j}^{x_{j+1}} F(x)\,dx \approx h\,F(x_{j-1/2}) \quad (2)$$

midpoint rule approximation.

exact u's

Now $u'\Big|_{x_j} \approx \frac{u_{j+1/2} - u_{j-1/2}}{h} \approx \ldots$ (short, centered differencing)

and so $u'\big|_{x_j} \approx \dfrac{U_{j+1/2} - U_{j-1/2}}{h^2}$

Similarly $u'\big|_{x_{j-1}} \approx \dfrac{U_{j-1/2} - U_{j-3/2}}{h^2}$

So (2) can be further approximated by the discrete scheme

$$-a(x_j)\,\frac{U_{j+1/2} - U_{j-1/2}}{h^2} + a(x_{j-1})\,\frac{U_{j-1/2} - U_{j-3/2}}{h^2} + U_{j-1/2} = h\,F(x_{j-1/2}).$$

Note that if $a(x) \equiv 1$ and we divide the equations above by h, we have the same discrete equations as we had for the FD approximation. Note that the linear system for the discretization is symmetric (desireable). The standard FD approach does not lead to a symmetric linear system as shown on the Day 10 notes, page 2.

Math 405/607, Fall 2014, Day 11

Midterm next Thursday, October 16, in class. The test will be 60 minutes and the class will end at the end of the test. Material includes:

- General ideas of discretization, iteration and convergence (two types)
- FP accuracy, matrix norms, condition number
- Bisection, Newton, vector Newton methods.
- Interpolation and accuracy.
- Spectral methods and von Neumann analysis.
- Quadrature and accuracy; adaptive methods
- Finite Difference approximations of derivatives (Taylor series).
- Finite Difference Methods for BVPs: Stability & consistency, Lax equivalence Theorem, implementing boundary conditions.
- will not include latest material on collocation methods.

Format: 5 questions each worth 5 marks, total 25.

4 questions part A

1 question part B (choice)

Consider the stability of the collocation scheme discussed last time, applied to the problem

$$-u'' + u = f(x)$$
$$u(x) \text{ 1-periodic}$$

written as a system $\underline{S} = (u, v)$

$$u' = v$$
$$v' = u - f.$$

We have discrete values $\underline{S} = (\underline{U}, \underline{V})$ on a grid of N points, equally spaced with spacing h.

Discrete equations:

$$\left(\frac{3}{2h} + \frac{h}{8}\right)(U_{j+1} - U_j) - \frac{3}{4}(V_j + V_{j+1}) = h(f_1 - f_0)/8$$
$$\left(\frac{3}{2h} + \frac{h}{8}\right)(V_{j+1} - V_j) - \frac{3}{4}(U_j + U_{j+1}) = \frac{1}{4}(f_1 - f_0) - f_{1/2}. \quad (1)$$

Can verify directly, with Taylor series, that these equations are consistent with $O(h^4)$ truncation error, but we did it already for the general case. We can write

$$U_j = \sum_{\alpha=0}^{N-1} \hat{U}_\alpha e^{2\pi i j \alpha / N}$$
$$V_j = \sum_{\alpha=0}^{N-1} \hat{V}_\alpha e^{2\pi i j \alpha / N}.$$

[DFT, Von Neumann analysis] to get us part of the way there.

Recall $\|\underline{U}\|_2 = \|\hat{\underline{U}}\|_2$, $\|\underline{V}\|_2 = \|\hat{\underline{V}}\|_2$

So $\|\underline{S}\|_2^2 = \|\underline{U}\|_2^2 + \|\underline{V}\|_2^2 = \|\hat{\underline{U}}\|_2^2 + \|\hat{\underline{V}}\|_2^2$

The Transforms partially diagonalize (1)

$$\left(\frac{3}{2h} + \frac{h}{8}\right)\left(e^{2\pi i\alpha/N} - 1\right)\hat{U}_\alpha - \frac{3}{4}\left(e^{2\pi i\alpha/N} + 1\right)\hat{V}_\alpha = O(h$$

$$\left(\frac{3}{2h} + \frac{h}{8}\right)\left(e^{2\pi i\alpha/N} - 1\right)\hat{V}_\alpha - \frac{3}{4}\left(e^{2\pi i\alpha/N} + 1\right)\hat{U}_\alpha = O(h^4$$

Thus we need to show that the 2×2 matrices $A_{\alpha,h}$

$$\|A_{\alpha,h}^{-1}\|_2 < C \qquad (2).$$

for some constant C independent of α and h to show l_2 stability and fourth order convergen of the scheme in l_2 norm, where

$$A_{\alpha,h} = \begin{pmatrix} \left(\frac{3}{2h} + \frac{h}{8}\right)\left(e_{\alpha,h} - 1\right) & -\frac{3}{4}\left(e_{\alpha,h} + 1\right) \\ -\frac{3}{4}\left(e_{\alpha,h} + 1\right) & \left(\frac{3}{2h} + \frac{h}{8}\right)\left(e_{\alpha,h} - 1\right) \end{pmatrix}.$$

and $e_{\alpha,h} = e^{2\pi i\alpha h}$. We can make analytic progress on this as follows.

$$A_{\alpha,h} = e^{\pi i\alpha h}\begin{pmatrix} \left(\frac{3}{h} + \frac{h}{4}\right) i\sin\pi\alpha h & -\frac{3}{2}\cos\pi\alpha h \\ -\frac{3}{2}\cos\pi\alpha h & \left(\frac{3}{h} + \frac{h}{4}\right) i\sin\pi\alpha h \end{pmatrix}$$

The algebra to get the form above corresponds to the idea that the discretization (1) is really centred at the midpoint of the subinterval.

Note that $\alpha = 0, \ldots N-1$ and $h = 1/N$, so

$$\pi\alpha h \in [0, \pi).$$

Thus if we consider

$$A_{\theta,h} = \begin{pmatrix} \left(\frac{3}{a} + \frac{h}{4}\right) i\sin\theta & -\frac{3}{2}\cos\theta \\ -\frac{3}{2}\cos\theta & \left(\frac{3}{a} + \frac{h}{4}\right) i\sin\theta \end{pmatrix}$$

and show $\| A^{-1}_{\theta,h} \|_2 < C$ for $\theta \in [0, \pi)$ and $h > 0$ then (2) follows. The eigenanalysis of A is

$$\lambda_{1,2} = \left(\frac{3}{a} + \frac{h}{4}\right) i\sin\theta \pm \frac{3}{2}\cos\theta.$$

$$\underline{v}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \quad \underline{v}_2 = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \quad T = \left[\underline{v}_1 \mid \underline{v}_2 \right].$$

$$A_{\theta,h} = T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} T^{-1}$$ ← diagonalization of A, encodes

$A \underline{v}_1 = \lambda_1 \underline{v}_1$

$A \underline{v}_2 = \lambda_2 \underline{v}_2$

and $\{\underline{v}_1, \underline{v}_2\}$ are a basis

$$A^{-1}_{\theta,h} = T \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix} T^{-1}.$$

$$\| A^{-1}_{\theta,h} \|_2 \le \| T \|_2 \max\left\{ \left|\frac{1}{\lambda_1}\right|, \left|\frac{1}{\lambda_2}\right| \right\} \| T^{-1} \|_2$$

Note that multiplication by T and T^{-1} corresponds to an orthonormal change of coordinate system, so $\| T \|_2 = \| T^{-1} \|_2 = 1$.

$$|\lambda_1|^2 = |\lambda_2|^2 = \frac{a}{4}\cos^2\theta + \left(\frac{3}{a} + \frac{a}{4}\right)^2 \sin^2\theta.$$

$$\geq \min\left(\frac{a}{4}, \left(\frac{3}{a} + \frac{a}{4}\right)^2\right)(\cos^2\theta + \sin^2\theta).$$

min of $\frac{3}{a} + \frac{a}{4}$ occurs at $a = \sqrt{12}$, value ≈ 1.73.

Thus $|\lambda_1| = |\lambda_2| \geq 3/2$.

$$\frac{1}{|\lambda_1|} = \frac{1}{|\lambda_2|} \leq 2/3.$$

$\|A^{-1}_{\theta,a}\|_2 \leq 2/3$, we have finished the proof of the stability result.

Math 405, day 9 notes addition, Fall 2014

To investigate the truncation error of time stepping schemes, we need multivariate Taylor series. Recall scalar function Taylor series:

$$f(x) = \underbrace{f(a) + f'(a)(x-a)}_{\text{linear (tangent line) approximation } T_1(x)} + \underbrace{\frac{1}{2} f''(\xi_1)(x-a)^2}_{\text{error term}}$$

$$f(x) = \underbrace{f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2}_{\text{quadratic approximation } T_2(x)} + \frac{1}{6} f'''(\xi_2)(x-a)^3$$

$$f(x) = \underbrace{\sum_{j=0}^{n} \frac{f^{(j)}(a)}{j!}(x-a)^j}_{T_n(x)} + \frac{f^{(n+1)}(\xi_n)}{(n+1)!}(x-a)^{n+1}.$$

where equality holds for some $\xi_n \in (a, x)$.

Note that these expressions lead to the estimates

$$|f(x) - T_n(x)| \le C_{n+1} |x-a|^{n+1} \qquad C_{n+1} = \frac{K_{n+1}}{(n+1)!}$$

as long as $f \in C_{n+1}$

Note also that $T_n(x)$ is the n'th order polynomial in x chosen so that the value of $T_n(a)$ matches $f(a)$, and $T_n^{(j)}(a)$ matches $f^{(j)}(a)$ for derivatives $j = 1, \dots n$.

We can construct polynomial approximations for functions of two (or more) variables in the same way. Consider $f(x,y)$ near (a,b).

$$f(x,y) \approx T_1(x,y) := f(a,b) + \frac{\partial f}{\partial x}(a,b)(x-a) + \frac{\partial f}{\partial y}(a,b)(y-b). \quad (1).$$

Note that $\frac{\partial T_1}{\partial x}(a,b) = \frac{\partial f}{\partial x}(a,b)$

and $\frac{\partial T}{\partial y}(a,b) = \frac{\partial f}{\partial y}(a,b)$.

(1) is the tangent plane approximation of f based at (a,b).

Higher order

$$f(x,y) \approx T_2(x,y) := f + f_x(x-a) + f_y(y-b) + \frac{1}{2} f_{xx}(x-a)^2 + f_{xy}(x-a)(y-b) + \frac{f_{yy}}{2}(y-b)^2$$

where above, f and its derivatives are evaluated at (a,b).

$$f(x,y) \approx T_3(x,y) = T_2(x,y) + \frac{1}{6} f_{xxx}(x-a)^3 + \frac{1}{6} f_{yyy}(y-b)^3$$

$$+ \frac{f_{xyy}}{2}(x-a)(y-b)^2 + \frac{f_{xxy}}{2}(x-a)^2(y-b).$$

General form

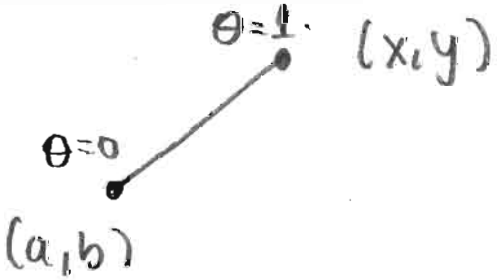
$$T_n(x) = \sum_{j=0}^{n} \sum_{i=0}^{j} \frac{1}{i!(j-i)!} \frac{\partial^j f}{\partial x^i \partial y^{j-i}}(a,b)(x-a)^i(y-b)^{j-i}$$

Theorem: If $f \in C_{n+1}$ then

$$|f - T_n(x)| \le C_{n+1} \|(x,y) - (a,b)\|^{n+1}.$$

where C_{n+1} depends on the size of the $n+1$ derivatives of f.

Proof: ($n=1$ case).



Consider the line segment between (a,b) and (x,y) parametrized by θ as shown. Let

$$g(\theta) = f(a + \theta(x-a),\ b + \theta(y-b))$$

and use scalar Taylor linear approximation

$$f(x,y) = g(1) = \underbrace{g(0)}_{f(a,b)} + g'(0) + \frac{1}{2} g''(\xi) \qquad (2)$$

for some $\xi \in (0,1)$.

$$g'(\theta) = \frac{\partial f}{\partial x}(a+\theta(x-a), b+\theta(y-b)) \cdot (x-a) + \frac{\partial f}{\partial y}(a+\theta(x-a), b+\theta(y-b)) \cdot (y-a).$$

$$g''(\theta) = f_{xx}(x-a)^2 + f_{xy}(x-a)(y-b) + f_{xy}(x-a)(y-b) + f_{yy}(y-b)^2$$

$$g'(0) = \frac{\partial f}{\partial x}(a,b)\cdot(x-a) + \frac{\partial f}{\partial y}(a,b)\cdot(y-b)$$

Thus $g(0) + g'(0) = T_1(x,y)$ and from (2)

$$|f(x,y) - T_1(x,y)| = |g''(\xi)|/2. \qquad (3).$$

$$\leq \frac{1}{2} K_2 \left\{ (x-a)^2 + (y-b)^2 + 2|(x-a)(y-b)| \right\}.$$

where $K_2 = \max_{(x,y)} \{ |f_{xx}|, |f_{xy}|, |f_{yy}| \}$.

Since $2|(x-a)(y-b)| < (x-a)^2 + (y-b)^2$

[follows from $(|\alpha| - |\beta|)^2 \geq 0$.], (3) becomes

$$|f(x,y) - T_1(x,y)| \leq K_2 \left\{ (x-a)^2 + (y-b)^2 \right\} \quad \checkmark.$$

Math 405/607, Fall 2014, Day 16 Notes

Let us now turn to the approximation of PDE problems. We will consider the following problems over the next few weeks:

(A) Parabolic problem $u(x,t)$

$$u_t = u_{xx} - u + f(x,t).$$

(B) Model hyperbolic problem $u(x,t)$

$$u_t + u_x = 0.$$

(C) Wave equation $u(x,t)$

$$u_{tt} - u_{xx} = 0.$$

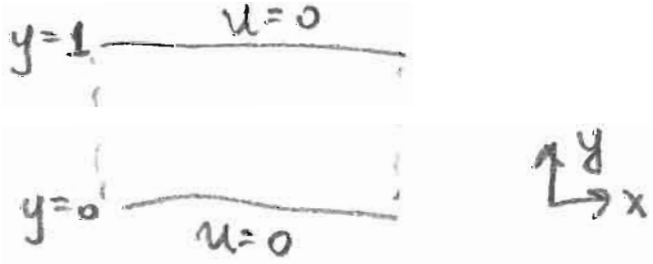
(D) Poisson Problem $u(x,y)$.

$$u_{xx} + u_{yy} = f(x,y).$$

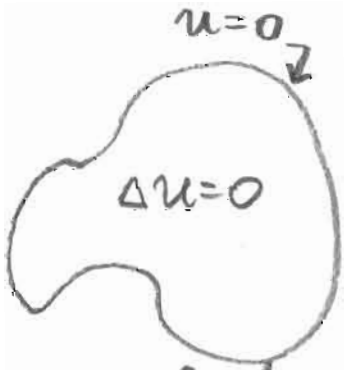
in a rectangular domain

$x \in [0,1]$, $y \in [0,1]$

u 1-periodic in x. OR



suitable for FD and spectral methods



general domains suitable for FE methods.

(A) $u_t = u_{xx} - u + f(x,t)$.

u, f 1-periodic in x for every t

$u(x,0) = u_0(x)$ given (initial conditions).

Note: We could replace x-periodicity by Dirichlet, Neumann, or Robin boundary conditions.

Note: If $f(x)$ does not depend on t, then the steady state solution of (A) is our old boundary value problem

$$-u_{xx} + u = f(x).$$

Note: The exact solution of (A) can be represented as a Fourier series:

$$u(x,t) = \sum_{\alpha=-\infty}^{\infty} \hat{u}_\alpha(t)\, e^{2\pi i \alpha x}$$

with $\hat{u}_\alpha(t) = \widehat{u_{0,\alpha}}\, e^{\lambda_\alpha t} + \int_0^t \hat{f}_\alpha(t)\, e^{\lambda_\alpha (t-s)}\, ds$

↑ Transform of the initial data.

with $\lambda_\alpha = -4\pi^2\alpha^2 - 1$.

Qualitatively, high wave number (oscillatory) components of the solution decay quickly.

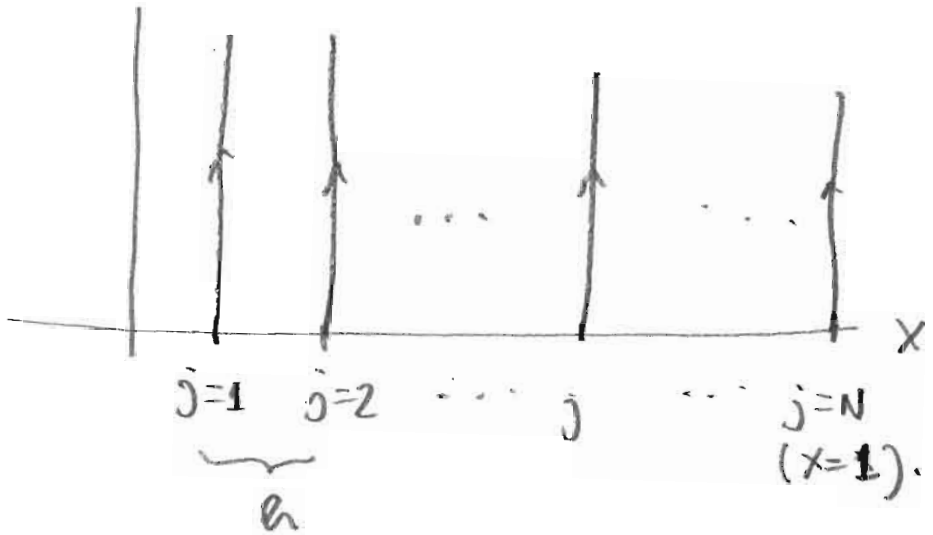
Our first step in the discretization of problem (A) will be a semi-discretization (Method of lines). We'll discretize in space, but leave time continuous.

$$U_j(t) \approx u(jh, t) \qquad j = 1, \ldots N.$$
$$h = 1/N.$$

Approximate the spatial derivatives with finite derivatives, keep the time derivatives exact:

$$\frac{dU_j}{dt} = D_2 U_j - U_j + f(jh, t). \qquad (1)$$

This is called the Method of lines because the approximation is done on lines in (x,t) space.



(1) is a first order ODE system. A good first step for small scale time dependent PDE problems is to discretize in space only

(MOL) and put the resulting ODE system (1) into a black box ODE solver.

Note: Theory can show that $\underline{U}(t)$ converges with second order to $\underline{u}(t)$. That is

$$\| \underline{U}(t) - \underline{u}(t) \|_\infty \leq C h^2.$$

For fixed h, if the ODE system is approximated by a time stepping method with M fixed time steps k sufficiently small to time $T = Mk$, and the method is p'th order, then

$$\| \underline{U}^M - \underline{U}(T) \|_\infty \leq K k^p.$$

However, $k \leq k_{min}(h)$ for this to occur as we shall see. Together, we obtain

$$\| \underline{U}^M - \underline{u}(t) \|_\infty \leq C h^2 + K k^p.$$

↑ fully discrete solution

↑ exact solution

But the condition $k \leq k_{min}(h)$ is a big issue. Let's examine that further. Return to (1):

$$\frac{dU_j}{dt} = D_2 U_j - U_j + F(jh, t).$$

We can write $\underline{U}(t)$ in terms of its DFT $\hat{\underline{U}}(t)$ and get

$$\frac{d\hat{U}_\alpha}{dt} = \lambda_\alpha \hat{U}_\alpha + \vec{F}_\alpha(t).$$

↙ Symbol of D_2.

with $\lambda_\alpha = \dfrac{2(\cos(2\pi\alpha h) - 1)}{h^2} - 1.$

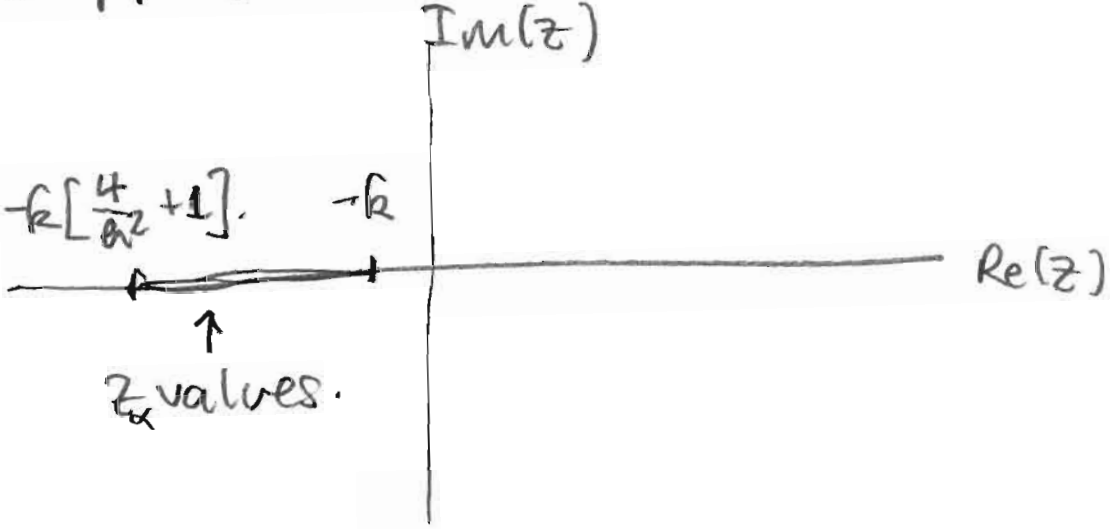
Consider the case $\vec{F}_\alpha(t) \equiv 0$. In this case,

$$\hat{U}_\alpha(t) = \hat{U}_\alpha(0)\, e^{\lambda_\alpha t}. \qquad (2)$$

Since λ_α is real and < 0 for every α, solutions (2) decay. In order for the fully discrete solutions to decay also,

$$z = k\lambda_\alpha$$

should be in the stability region of the time stepping scheme for every α.



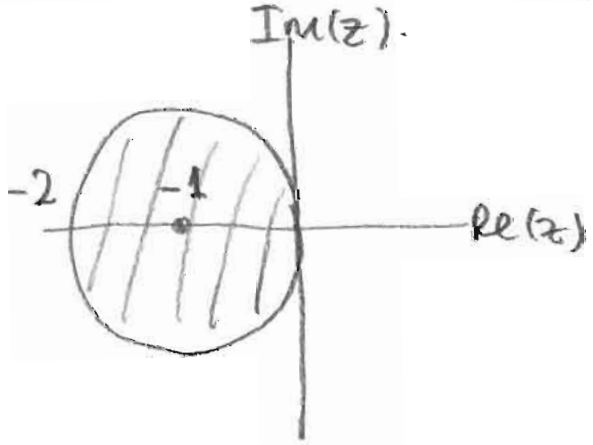
$\lambda_\alpha \in \left[-\frac{4}{h^2} - 1, -1\right]$. This makes (1) a stiff problem.

If the time stepping scheme is L-stable, all components of the solution decay as they should without time step restriction ✓

Consider now the use of a time-stepping scheme that is not L-stable. For example, consider FE applied to (1):

$$U_j^{n+1} = U_j^n + \frac{k}{h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right) + F(jh, nk) - k\,U_j^n.$$

Recall the stability region for FE:



For $k\lambda$ to be in the stability region, we need

$$k\left[\frac{4}{h^2} + 1\right] < 2$$

or $$k < \frac{2h^2}{4+h^2} \approx \frac{h^2}{2}$$

If we used Improved Euler, we would have the same stability restriction on time step size. In general, we would want to choose h & k independently based on accuracy

requirements. Thus, we would want for this problem a time-stepping scheme that contained the entire negative real axis in its stability region. L-stable schemes guarantee this.

Test problem:

Consider $u(x,t) = \sin t\, e^{\cos x}$. This solves (1) with

$$f(x,t) = e^{\cos x}\left(\cos t + \sin t\left(1 - \sin^2 x + \cos x\right)\right)$$

and $u_0(x) \equiv 0$.

This is solved out to time 1 with FE and BE time stepping in the posted MATLAB codes,

fe_diffusion.m and be_diffusion.m.

For parabolic (diffusion) type problems, it can be important to avoid stability restrictions by using L-stable schemes. A-stable schemes have additional desireable properties.

(B) $u_t + u_x = 0 \qquad u(x,0) = u_0(x)$ given.

Solution is $u(x,t) = u_0(\underbrace{x-t}_{\text{argument taken mod with the periodicity.}})$

Thus, we call (B) a "one way wave equation" since solutions move to the right with speed 1.

If we consider Fourier series solutions to this problem,

$$\hat{u}_\alpha(t) = \widehat{u_0}_{,\alpha}\, e^{\lambda_\alpha t}$$

where $\lambda_\alpha = -2\pi i \alpha$

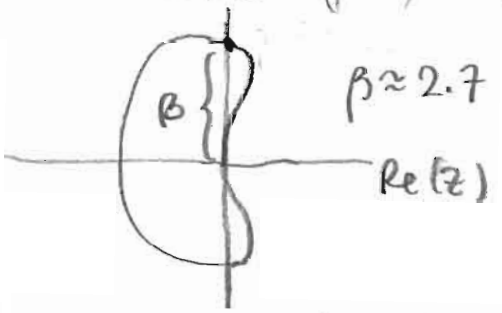
we have $|\hat{u}_\alpha(t)| = |\widehat{u_0}_{,\alpha}|$, i.e. the Fourier series coefficients do not vary in magnitude with time.

We could discretize (B) in space, but not time:

$$\dot{\underline{U}} = -D_1 \underline{U}. \qquad (3)$$

The symbol of D_1 is $i\sin(2\pi i \alpha h)/h$, so the discrete problem also has pure imaginary eigenvalues.

Higher order Runge-Kutta methods are often good, "safe" choices for time stepping for (3). They are explicit, and RK methods of high order can contain a segment of the imaginary axis in their stability region. For example, RK4 (standard 4 stage, 4th order RK).



So all components of the solution (3) will decay

if $k < \beta h$. (4).

↑

This stability restriction is called a CFL condition (Courant - Friedrich - Levy). Sometimes the term is applied to stability conditions for parabolic problems.

Note: For higher order time stepping schemes, the boundary of the stability region is very close to the imaginary axis for z near zero. Thus, small α values do not decay (or grow) very much. Large α values should have small magnitudes if the solution is smooth, so handling them correctly is not so important. However, if they grow geometrically (that is, if they correspond to z values outside of the stability region) then they will limit the accuracy for long time computations. Thus, it is considered better to have them decay in time.

Note: There are specialized methods for (3) and the two-way wave equation (C) that do not come from the MOL approach. Some are considered in assignment #4.

Note: (3) is on the borderline of being considered a stiff problem. The stability restriction (4) is an order of magnitude better than that for

parabolic problems. It is generally considered reasonable to use explicit time stepping for these problems. In addition, the linear algebra problem from implicit time stepping of hyperbolic problems is less amenable to iterative solution in higher dimensional settings.

This problem has some similarities with the undamped spring

$$\ddot{u} + u = 0.$$

I will apply several of MATLAB's time stepping routines to this problem, then apply ode45 to (3) with

$$u_0(x) = e^{\cos x}$$

in the 2π-periodic setting. Codes

undamped.m
undamped_call.m
and oneway.m

are posted.

Math 405/607, Fall 2014, Day 17 Notes

So far in the class, we have concentrated mostly on quantitative aspects of errors from numerical approximation. We have considered the size of errors from convergent

- iterative procedures
- spatial discretization
- time stepping schemes.

However, we have also considered the qualitative behaviour of some schemes, and I'd like to highlight some important ideas.

(1) $\ddot{u} + u = 0$, $\dot{u} = v$, $\dot{v} = -u$ has solutions such that

$$(\dot{u})^2 + u^2$$

is constant in time (it is the scaled total energy, kinetic plus potential, of an unpumped spring-mass system).
We found (assignment #3, A5) that TR time stepping maintained this property exactly. The TR solutions are not exact (they have oscillation period $2\pi + O(k^2)$) but they do preserve the energy invariance exactly. Sometimes, it is important to preserve some invariants exactly, and it is often possible to do so (with some work).

There are large computations done of coupled undamped oscillators, modelling molecular dynamics. For these large problems, it is not feasible to solve the large, implicit, system that would arise from TR (or other implicit) time stepping. Non standard, explicit time stepping techniques are used (like assignment #4, A4) that have an invariant that approximates the true one. In this context, such schemes are called symplectic methods.

(2) $u_t = u_{xx}$ $\quad u(x,0) = u_0(x)$ given.

$u(x,t)$ 1-periodic in x.

This problem has a maximum principle, that is

$$u(x,t) \leq \max_x u_0(x)$$

for all x and all $t \geq 0$. It also has a minimum principle

$$u(x,t) \geq \min_x u_0(x)$$

(obtained by considering $-u$). These can be extended to include Dirichlet boundary conditions.

Fully discrete schemes can be constructed so that they also have this property:

$$U_j^n \le \max_j U_0(jh).$$

You will see two such schemes in assignment #4, A1.

Let us return to problem (B), MOL discretization

$$\dot{\underline{U}} = -D_1 \underline{U} \qquad (1)$$

eigenvalues $\lambda_\alpha = -i\sin(2\pi\alpha h)/h$. Suppose we solved (1) exactly – there would still be errors in the solution due to the spatial discretization. The approximation in the α Fourier component is

$$\widehat{U_0}_\alpha \underbrace{e^{i2\pi\alpha x} e^{-2\pi i\alpha t}}_{e^{2\pi i\alpha(x-t)}}$$

wave moving to the right with speed 1

by $\widehat{U_0}_{,\alpha} e^{2\pi i j\alpha h} \cdot e^{-i[\sin(2\pi\alpha h)/h]t}$

taking $jh = x_j$ this has the form

$$e^{2\pi i\alpha\left(x_j - \frac{\sin(2\pi\alpha h)}{2\pi h\cdot\alpha} t\right)}$$

This is a wave travelling to the right with speed $\dfrac{\sin(2\pi\alpha h)}{2\pi h\cdot\alpha}$.

Note that the wave speed in the spatial approximation (MOL) depends on the wave number. Considering fixed α as $h \to 0$,

[$\sin x \approx x - x^3/6$ for x small]

$$\frac{\sin(2\pi\alpha h)}{2\pi\alpha h} \to 1 - \frac{1}{6}(2\pi\alpha h)^2 + O(h^4).$$

↑ exact wave speed

↑ dominant error, a dispersive effect.

Higher wavenumber components move slower... Controlling the effects of numerical dispersion is important in long time simulations of wave propagation.

Summary: While it is important to have a convergent scheme (quantitatively accurate), it might be equally important to preserve, or at least control, some qualitative behaviors of the solution. For certain specific types of problems, specialized types of schemes can be built to do exactly that.

Let us return to the non-specialized path, considering

(C) Wave equation $u(x,t)$

$$u_{tt} - u_{xx} = 0,$$

$$u(x,0) = u_0(x), \qquad u_t(x,0) = v_0(x) \quad \text{given}$$

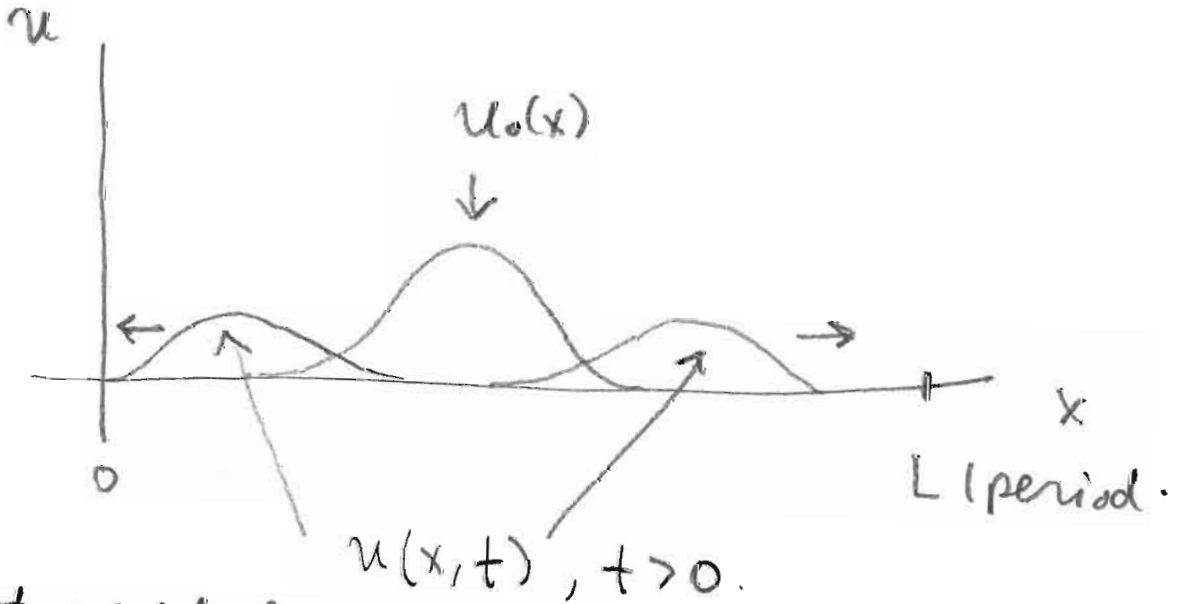
It is known that all solutions of (C) can be written in the form

$$u(x,t) = F(x-t) + g(x+t).$$

For some functions of one variable $F(\cdot)$ and $g(\cdot)$ that depend on the initial and boundary value data. For example, if $V_0(x) \equiv 0$ and the problem is taken in the periodic setting,

$$u(x,t) = \tfrac{1}{2}(u_0(x-t) + u_0(x+t)).$$

In words, the initial values $u_0(x)$ become waves moving to the right & left.



A straight-forward discretization begins with the MoL, $\underline{U}(t)$ solving

$$\ddot{\underline{U}} = D_2 \underline{U}.$$

To use standard time stepping techniques, we convert this to a first order system, introducing $\underline{V} = \dot{\underline{U}}$:

$$\left.\begin{aligned} \dot{\underline{U}} &= \underline{V} & U_j(0) &= u_0(jh) \\ \dot{\underline{V}} &= D_2 \underline{U}. & V_j(0) &= v_0(jh). \end{aligned}\right\} \quad (2)$$

The system can be consolidated to

$$\underline{\mathcal{U}} = \begin{bmatrix} \underline{U} \\ \underline{V} \end{bmatrix}.$$

and $$\dot{\underline{\mathcal{U}}} = \mathcal{A}\,\underline{\mathcal{U}}$$

where $\mathcal{A}$ is a $2N \times 2N$ matrix with block form

$$\mathcal{A} = \begin{bmatrix} 0 & \mathbb{I} \\ D_2 & 0 \end{bmatrix}.$$

We are interested in the eigenvalues of $\mathcal{A}$. I am going to spare us some of the work by saying the eigenvectors have the form

$$\underline{U} = \begin{bmatrix} a\, e^{2\pi i j \alpha h} \\ b\, e^{2\pi i j \alpha h} \end{bmatrix} \qquad (a, b \text{ depending on } \alpha \text{ \& } h)$$

which is an eigen vector of $\mathcal{A}$ iff $\begin{bmatrix} a \\ b \end{bmatrix}$ is an eigenvector of

$$\begin{bmatrix} 0 & 1 \\ \dfrac{2(\cos(2\pi\alpha h) - 1)}{h^2} & 0 \end{bmatrix} \qquad (3)$$

↑ eigenvalue (symbol) of D_2.

The eigenvectors are

$$\lambda_\alpha = \pm \sqrt{\frac{2(\cos(2\pi\alpha h) - 1)}{h^2}}$$

Thus the eigenvalues are pure imaginary, with magnitude up to size $2/h$. This gives insight into the CFL step size restriction needed for stable computation with appropriate explicit time stepping methods.

Notes: (i) The eigenvectors of (3) are not orthogonal. This makes the complete stability analysis of this discretization interesting.

(ii) A specialized scheme, not MOL, is considered in assignment #4, A5.

(iii) The wave equation can also be written as a first order system in a different way, introducing $V = u_t$ and $W = u_x$.

$$W_t = u_x \quad \text{equality of mixed partials.}$$

$$V_t = W_x \quad \text{wave equation}$$

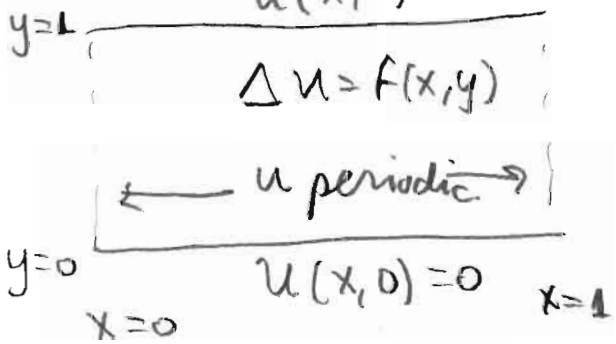
This system, when discretized appropriately, has slightly better stability properties than (2).

Math 405/607, Day 18, Fall 2014

Let us now consider our final PDE problem of the course, for $u(x,y)$ in a bounded domain Ω

(D) $\underbrace{\Delta u}_{u_{xx}+u_{yy}} = f(x,y)$ with $u=0$ on the boundary of Ω.

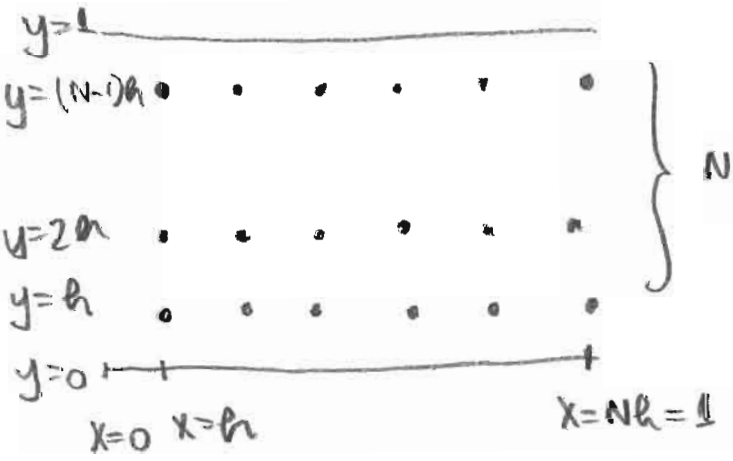
Model domain, unit square, u periodic in x.



general domain

↑ discretized with the finite element method, next topic.

Discretize this problem, using $N(N-1)$ values on a 2D grid; with $h = 1/N$.



We have $U_{ij} \approx u(ih, jh)$

$i = 1, \cdots N, \quad j = 1, \cdots N-1.$

To approximate (D) on this grid, we take

$$D_{2,x} U_{ij} + D_{2,y} U_{ij} = F_{ij} = F(ih, jh).$$

Now, at interior points this reads.

$$\frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{h^2} = F_{ij}$$

↑

$D_{2,x}$

or combining

$$\frac{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4U_{ij}}{h^2} = F_{ij} \quad (1)$$

↗

this defines

$\Delta_h U_{ij}$

Note: We do not necessarily have to use $\Delta x = \Delta y$.

(1) represents a linear system for the $N(N-1)$ unknowns U_{ij}. To implement it, we have to order the U_{ij} values into a vector of length $N(N-1)$. A typical way to do this, called lexicographical order, by x coordinate rows in increasing y.

$$\underline{U} = \left[U_{1,1} \;\; U_{2,1} \cdots U_{N,1} \;,\; U_{1,2},\; U_{2,2} \cdots U_{N,2}, \cdots U_{N,N-1} \right]$$

or $U_\ell = U_{i,j}$

where $\ell = (i-1)\cdot N + j$ and

$$\begin{cases} i = \text{floor}\left(\frac{\ell-1}{N}\right) + 1. \\ j = \ell - (i-1)N. \end{cases}$$

With these maps, (1) can be converted into the form

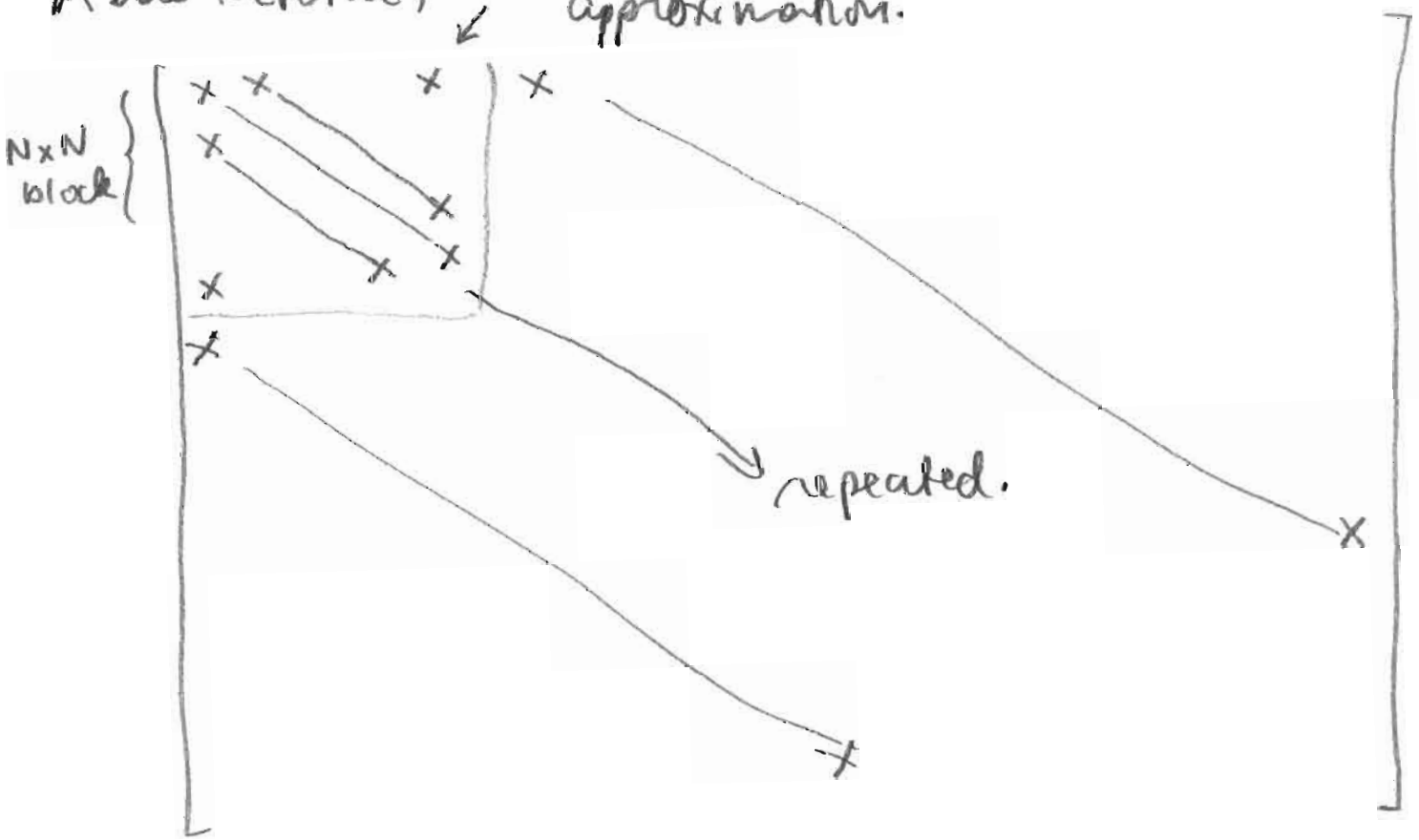
$$[A]\underline{U} = \underline{F}$$

Note: the result converges with $O(h^2)$ errors.

$[A]$: $N(N-1) \times N(N-1)$ sparse matrix.

$\underline{U}$, $\underline{F}$: index ℓ, $N(N-1)$ sized vectors.

$[A]$ has the form, ← derivative approximation.



With this ordering, the thick band around the diagonal of width $2N+1$ fills in with nonzero entries during Gaussian elimination. The resulting operation count is $O(N^4)$.

Let's compare operation counts for 1D versus 2D boundary value problems. N is a good way to compare since $N = O(1/h)$ measures the accuracy in both cases.

	1D	2D.
unknowns	$O(N)$	$O(N^2)$
direct solve (work)	$O(N)$	$O(N^4)$.
direct solve (storage)	$O(N)$	$O(N^3)$.
error	$O(\frac{1}{N^2})$	$O(\frac{1}{N^2})$

So in order to reduce the error by a factor of 4 in 1D you need to do 2 times more work but in 2D using the approach above you need to do 16 times more work.

4 times because $N^2 \rightarrow (2N)^2 = 4N^2$ unknowns.

another 4 times because of inefficiency in the solver.

Notes: The number of unknowns is even larger (N^3) in 3D and a sparse solver using lexicographical ordering is even less efficient.

However, it is still possible to do 2D and small 3D problems quickly on modern personal machines, using direct solvers.

Large 3D problems, or 2D & 3D problems that are solved repetitively, i.e. in an optimization process or inverse problem, require fast solvers.

Some fast solvers:

- (specific to this problem). Take the DFT in each row. The x-components in each row, gathered as a vector, each solve a (different) tridiagonal system with $N-1$ unknowns.

  Computational cost $O(N^2 \log N)$ [almost optimal].

- (for symmetric, definite systems [eigenvalues all the same sign]) Use the conjugate gradient method. Each iteration requires a multiplication by A ($O(N^2)$ operations here). Iteration error is reduced to size ε in

$$\leq \frac{1}{2}\sqrt{K}\,\ln\left(\frac{2}{\varepsilon}\right) + 1$$

iterations, where K is the condition number of A. As in 1D, $K = O(1/h^2) = O(N^2)$ so

This method converges to the desired tolerance with $O(N^3 \log(\frac{1}{\varepsilon}))$ cost.

This method is quite general, and some systems can be made to have smaller condition number with preconditioning.

This is a very general approach that can be extended (less efficiently) to non-(symmetric, definite) matrices, i.e. GMRES.

- (requires specific problem structure) Nested dissection, a clever reordering of the unknowns to reduce computational cost of the direct solve $O(N^2 \log N)$ operations.
- Multigrid methods. Very problem specific to get to work, but fast $O(N^2 \log(\frac{1}{\varepsilon}))$. Sometimes difficult to get to be convergent, but often still useful as preconditioners.

We will continue for the next few lectures to talk about MG methods, on a 1D model problem. I will also post notes on the conjugate gradient method, for your reference.

Let's go back to our old friend, $u(x)$, 1-periodic

$$-u'' + u = f(x)$$

discretized

$$-D_2 U_j + U_j = F_j := f(jh)$$

$$A\underline{U} = \underline{F} \qquad (2)$$

Remember, we have a fantastic, sparse direct solver for this problem, we are just using it as an easy framework for MG methods, which are efficient solvers in 2D & 3D.

The basis for MG methods are pointwise relaxation methods, which are inefficient as solvers but have one desireable property.

Iterations $\underline{U}^{(m)}$ that approximate $\underline{U}$, the exact solution of (2). Divide $\mathbb{A}$ into its diagonal part $\mathbb{D}$ and its lower & upper off diagonal parts $\mathbb{L}$ & $\mathbb{U}$.

$$\mathbb{A} = \mathbb{D} + \mathbb{L} + \mathbb{U}.$$

Weighted Jacobi iterations are described by

$$\underline{U}^{(m+1)} = \omega \mathbb{D}^{-1} \left(\underline{F} - (\mathbb{L}+\mathbb{U}) \underline{U}^{(m)} \right) + (1-\omega) \underline{U}^{(m)}$$

for $0 < \omega < 1$. Written in terms of components and applied to (2), we have

$$U_j^{(m+1)} = \omega \left(\frac{h^2}{2+h^2} \right) \left(F_j + \left(U_{j+1}^{(m)} + U_{j-1}^{(m)} \right) / h^2 \right) + (1-\omega) U_j^{(m)} \qquad (3)$$

It takes $O(N)$ operations to do one iteration, looping through all j values. If $\omega = 1$, (3) represents an update of the U_j value to satisfy (2) if the U_{j-1} and U_{j+1} from the previous iteration are used.

Note that the exact solution $\underline{U}$ satisfies (3) when used for $\underline{U}^{(m+1)}$ & $\underline{U}^{(m)}$ values.

$$U_j = \omega \left(\frac{h^2}{2+h^2}\right)\left(F_j + (U_{j+1} + U_{j-1})/h^2\right) \quad (4).$$
$$+ (1-\omega)\, U_j$$

subtracting (4) from (3), and using the notation

$$\underline{E}^{(m)} = \underline{U}^{(m)} - \underline{U}$$

(the iteration error), we have

$$E_j^{(m+1)} = \frac{\omega}{2+h^2}\left(E_{j+1}^{(m)} + E_{j-1}^{(m)}\right) + (1-\omega)\, E_j^{(m)}. \quad (5)$$

In general, $\underline{E}^{(m+1)} = \left[-\omega\, \mathbb{D}^{-1}(\mathbb{L}+\mathbb{U}) + (1-\omega)\,\mathbb{I}\right]\underline{E}^{(m)}$.

We want $\underline{E}^{(m)} \to \underline{0}$ as $m \to \infty$, which will happen if all the eigenvalues of

$$\left[-\omega\, \mathbb{D}^{-1}(\mathbb{L}+\mathbb{U}) + (1-\omega)\mathbb{I}\right]$$

are less than 1 in absolute value. Considering (5) we can see that the DF vectors are the eigenvectors of. Consider

$$E_j^{(m)} = \lambda_\alpha^m\, e^{2\pi i \alpha j h} \quad (6)$$

in (5) to get

$$\lambda_\alpha = \frac{2\omega}{2+h^2}\cos(2\pi\alpha h) + (1-\omega)$$

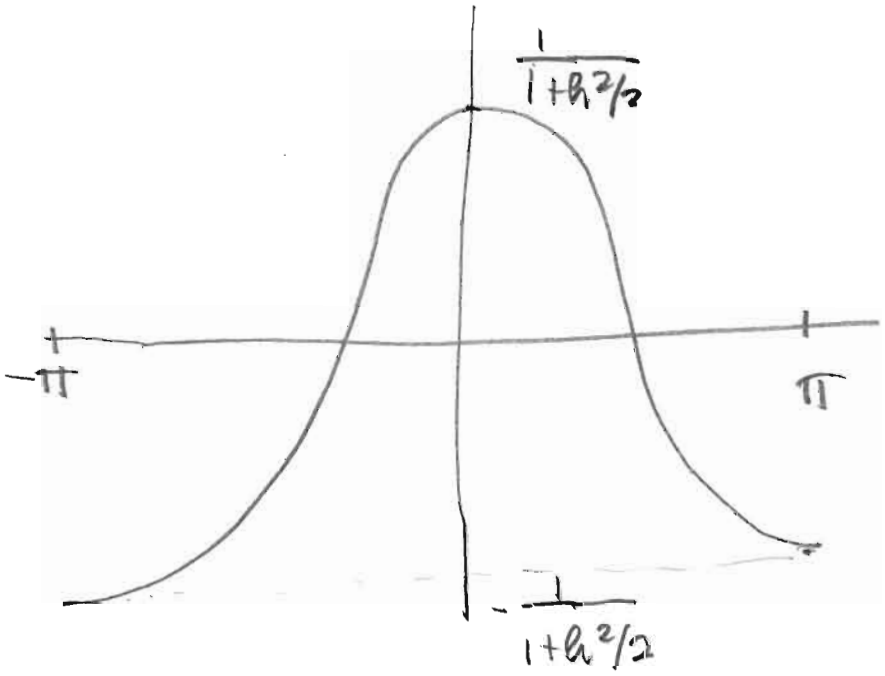
take as $\theta \in [-\pi, \pi]$.

$$= \omega\left[\frac{\cos(2\pi\alpha h)}{1 + h^2/2} - 1\right] + 1.$$

Consider $\omega = 1$, so

$$\lambda_\alpha = \frac{\cos(\theta)}{1+h^2/2}, \text{ clearly } |\lambda_\alpha| < 1.$$

We can plot $\lambda(\theta)$:



Considering (6), convergence is slow for $\theta \approx 0$ ($\alpha = 0$) and $\theta \approx \pm\pi$. ($\alpha = \pm N/2$). To get

$$\|\underline{E}^{(m)}\| \leq \varepsilon \, \|\underline{E}^{(0)}\|$$

we need M iterations such that

$$\left(\frac{1}{1+h^2/2}\right)^M \leq \varepsilon.$$

$$M \log(1+h^2/2) \leq \log \varepsilon$$

$$M \leq \frac{\log \varepsilon}{\log(1+h^2/2)} \approx \frac{\log \varepsilon}{h^2/2} = O(N^2 \log \varepsilon).$$

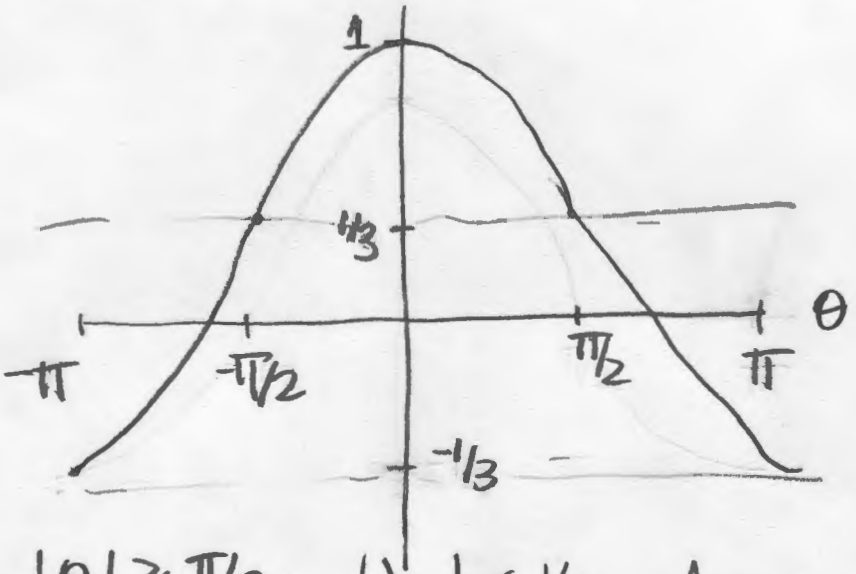
These iterations converge, but slowly... but they will still be useful for MG methods when we take $w > 1$, as we will see next week.

Math 405/607, Fall 2014, Day 19 Notes

Following on from Day 18 Notes, consider $\omega = 2/3$ as a weight in Jacobi iterations for our model problem.

$$\lambda_\alpha = \frac{2}{3}\left[\frac{\cos(\theta)}{1+h^2/2}\right] + 1/3.$$

Now $\max|\lambda_\alpha| = \frac{2/3}{1+h^2/2} + 1/3 > \frac{1}{1+h^2/2}$ so it seems doing this makes the iterations perform worse than before. However, if we consider $\lambda_\alpha(\theta)$ we see an important property

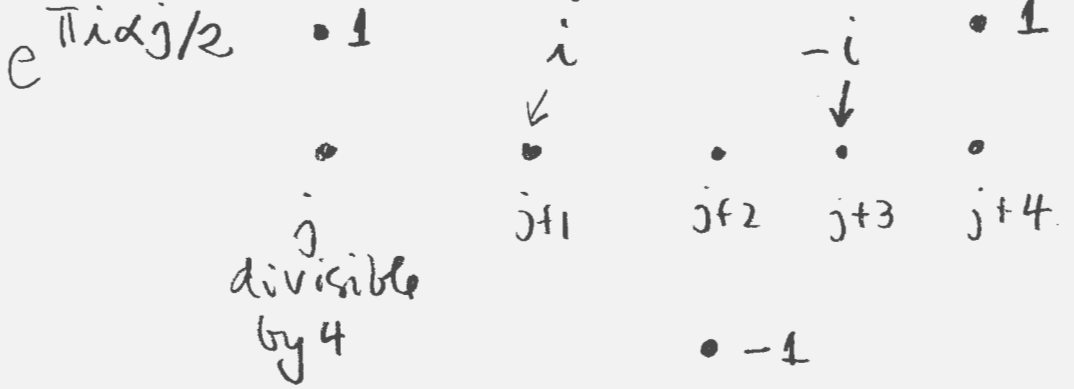


So for $|\theta| \geq \pi/2$, $|\lambda_\alpha| < 1/3$. Since $|\theta| > \pi/2$ corresponds to $\alpha h > 1/4$, these components of the solution are high wavenumber components on the grid.

$\theta = \pm\pi \Rightarrow \alpha h = 1/2$, corresponds to a DF vector ($\theta = 2\pi\alpha h$) $e^{\pi i \alpha j} = (-1)^j$, a vector that

alternates in sign between adjacent positions on the grid.

$\theta = +\pi/2 \Rightarrow \alpha h = 1/4$, corresponds to a DF vector



(and $\theta = -\pi/2$ its complex conjugate).

So with $w = 2/3$, after iterating m times with weighted Jacobi iterations, the error in the high wavenumber components ($\alpha h > 1/4$) is reduced by a factor of $(\frac{1}{3})^m$ in magnitude. We call such an iterative method a smoothing method, and

$$\mathbb{S} = \left[-\tfrac{2}{3} \mathbb{D}^{-1} (\mathbb{L} + \mathbb{U}) + \tfrac{1}{3} \mathbb{I} \right]$$

a smoothing operator. Note that if we $\pm \mathbb{D}$ here, we get the form

$$\mathbb{S} = \left[-\tfrac{2}{3} \mathbb{D}^{-1} \mathbb{A} + \mathbb{I} \right] \qquad (1).$$

Let's take an aside to talk about residual correction.

$$\underline{E}^{(m+1)} = S\,\underline{E}^{(m)} \leftarrow (2) \quad \underline{E}^{(m)} = \underline{U}^{(m)} - \underline{U}$$

($\underline{U}^{(m)}$: approximate; $\underline{U}$: exact.)

We compute $\underline{U}^{(m)}$ but we don't know $\underline{U}$ and so don't know $\underline{E}^{(m)}$ but (2) was still a useful theoretical result. If we know $\underline{U}^{(m)}$ we can determine that it is not exact by computing the residual

$$\underline{Q} = /\!A\,\underline{U}^{(m)} - \underline{F} \qquad \leftarrow /\!A\,\underline{U} = \underline{F} \text{ satisfied by the exact solution.}$$

$$= /\!A(\underline{U}^{(m)} - \underline{U}) - /\!A\,\underline{E}^{(m)}$$

so $\underline{E}^{(m)} = /\!A^{-1}\,\underline{Q}$, and we could get the exact solution $\underline{U}$ from $\underline{U}^{(m)}$ and its residual by

$$\underline{U} = \underline{U}^{(m)} - \underline{E}^{(m)} = \underline{U}^{(m)} - /\!A^{-1}\underline{Q}$$

$$= \underline{U}^{(m)} - /\!A^{-1}\,/\!A\,\underline{E}^{(m)}$$

subtract $\underline{U}$ from both sides and compare to (1) & (2).

$$\underline{0} = (\mathbb{I} - /\!A^{-1}\,/\!A)\,\underline{E}^{(m)}$$

↑ no error if exact $/\!A^{-1}$ is used to correct the residual.

We see that weighted Jacobi iterations are a residual correction method, using

$$\omega D^{-1} \text{ to approximate } A^{-1},$$

Jacobi iterations are great at getting the high wavenumber components of the solution correct (high wavenumber components of the error are reduced quickly).
The key idea of MG methods is that now, the remaining components of the error can be resolved on a coarse grid.

Start with $\underline{U}^{(0)} = \underline{0}$, so $\underline{E}^{(0)} = -\underline{U}$. Do a weighted Jacobi iteration to get $\underline{U}^{(*)}$, with $\underline{E}^{(*)}$ that satisfies

$$\underline{E}^* = S\,\underline{U}^{(0)}.$$

We have a residual on the fine grid of

$$A\underline{U}^* - \underline{F} = A\underline{E}^*$$

This residual is smooth, we can try and resolve it on a coarse grid ($2h$ spacing). Transfer (restrict) the residual to the coarse grid.

$$R A \underline{E}^*$$

R: $N/2$ by N matrix

Solve for the correction (exactly for now) on the coarse grid

$$I A_{2h}^{-1} I R I A \underline{E}^*$$

↑

N/2 by N/2

Then transfer (prolongate) the correction on the coarse grid back to the fine grid

$$I P\ I A_{2h}^{-1} I R I A\ \underline{E}^*$$

Make the correction

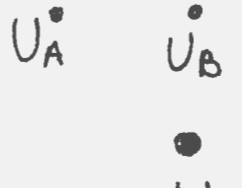
$$\underline{U}^{(**)} = \underline{U}^{(*)} - I P\ A_{2h}^{-1} I R I A\ \underline{E}^{(*)}$$

subtracting $\underline{U}$ from both sides, we have

$$\underline{E}^{(**)} = \left(\mathbb{I} - I P\ I A_{2h}^{-1} I R I A \right) \underline{E}^{(*)}$$

Notes: • This is a different residual correction method, with $I R^T A_{2h}^{-1} I R$ an approximation of $I A^{-1}$. Note that this is a complementary iteration to Jacobi, since it reduces the error in smooth (small wave number α) components.

• The restriction operator I consider is



fine grid.

coarse grid.

$$U_D = \frac{1}{4}(U_A + U_C) + \frac{1}{2} U_D.$$

We take $P = 2 R^T$. Note that R & P preserve the size of constant vectors. Depending on the way A_{2h} is defined and the dimension of the problem, P may have different scalings of R^T.

- After coarse grid correction, there is usually another smoothing sweep. We have:

$$\underline{E}^{(m+1)} = \underbrace{S\,(I - P A_{2h}^{-1} R A)\, S}_{M_2}\, \underline{E}^{(m)}. \qquad (3).$$

This is the 2-grid operator. It can be analyzed analytically with von-Neumann analysis, but I will show its properties numerically instead.

- Note that A_{2h}^{-1} represents solving a system with $1/2$ (1D), $1/4$ (2D), $1/8$ (3D) the number of unknowns as A_h^{-1}.
- Note that M_2 represents a residual reduction method

$$M_2 := I - Q A$$

(defines Q)

where Q is the underlying approximation of A^{-1}.

Solving for $\mathbb{Q}$ we obtain.

$$\mathbb{Q} = (\mathbb{I} - M_2)\mathbb{A}^{-1}.$$

- If we recursively used the 2-grid strategy on the system of equations in (3) symbolically represented by $\mathbb{A}_{2h}^{-1}$ we would then have error reduction of

$$\left.\begin{aligned} \underline{E}^{(m+1)} &= M_3 \underline{E}^{(m)} \\ \text{where } \mathbb{M}_3 &= \mathbb{S}(\mathbb{I} - \mathbb{P}\mathbb{Q}\mathbb{R}\mathbb{A})\mathbb{S} \end{aligned}\right\} (4)$$

$$\text{and } \mathbb{Q} = \left(\mathbb{I}_{2h} - S_{2h}\left(I_{2h} - P_{2h}\mathbb{A}_{4h}^{-1}R_{2h}A_{2h}\right)S_{2h}\right)\cdot \mathbb{A}_{2h}^{-1}.$$

Note that (4) is a theoretical tool, $\mathbb{A}_{2h}^{-1}$ is not needed in the method, only a solve with matrix $\mathbb{A}_{4h}$.

- We can continue to coarser and coarser grids recursively. The direct solve on the coarsest grid has negligible computational cost. The operation cost on all levels of smoothing, prolongation & restriction of one iteration is of order of the number of unknowns (geometric series). It can be shown that the eigenvalues of all M_j satisfy $|\lambda_j| < c < 1$ with c independent of j.

Thus in 1D, we have an

$$O(N|\log\varepsilon|)$$

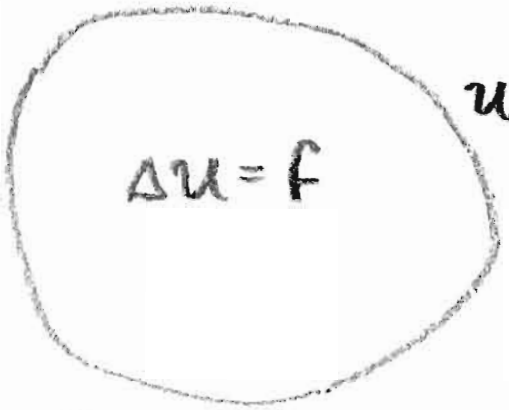
computational cost, in 2D

$$O(N^2|\log\varepsilon|), \quad 3D \quad O(N^3|\log\varepsilon|).$$

where ε is the desired precision in $\underline{U}$.
This is quite efficient compared to the direct solve with lexicographical ordering...

Math 405/607, Fall 2014, Day 20 Notes

Last problem to look at, (D) in a complex 2D domain:



A technique that can be used to approximate solutions to this problem is the Finite Element Method (FEM).

Let's go back to our old friend, the problem in 1D for $u(x)$, 1-periodic, that satisfies

$$-u'' + ru = f(x) \qquad [f \text{ given}]. \qquad (1)$$

to develop the fundamental ideas of the FEM in a simple setting.

If u satisfies (1), then we could multiply (1) by any smooth function $\varphi(x)$, integrate from 0 to 1, integrate the first term by parts to obtain:

$$\int_0^1 u'\varphi'\,dx + \int_0^1 u\varphi\,dx = \int_0^1 f(x)\varphi(x)\,dx \quad (2)$$

Defn If $u(x)$ satisfies (2) for every smooth function φ, we call u a weak solution of (1).

Notes: • Going from (1) $\Rightarrow$ (2) works for continuous, piecewise differentiable functions $\varphi(x)$ ✓.

• Weak solutions are unique, so if a weak solution has 2 continuous derivatives, it also satisfies (1) [a strong solution].

• Weak solutions are well defined even if f is not continuous (even if f is a distribution). Thus (2) is a generalization of (1).

The basic idea of the FEM is to look for approximate weak solutions $U(x)$ in a finite dimensional subspace of functions S_N spanned by N basis functions $\{\psi_j(x)\}_{j=1}^N$

We will look for approximate weak solutions of the form

$$U(x) = \sum_{j=1}^{N} U_j \psi_j(x) \qquad \text{(N unknowns)}$$

that satisfy (2) for every $\varphi \in S$, which will be true if

$$\int_0^1 U'(x)\,\psi_i'(x)\,dx + \int_0^1 U(x)\,\psi_i(x)\,dx = \int_0^1 f(x)\,\psi_i(x)\,dx \qquad (3)$$

for $i = 1, \cdots N$.

Note: (3) can also be obtained from a functional optimization approach. The solution of (1) minimizes

$$I(u) = \int_0^1 \{(u')^2 + u^2 - fu\}\,dx$$

and (3) represents the minimization of $I(U)$ over $U \in S$.

Continuing with (3), we obtain a linear system for the unknown U_j values in the form:

$$\mathbb{A}\,\underline{U} = \underline{F}$$

where $\mathbb{A} = \mathbb{K} + \mathbb{M}$ and

$$K_{ij} = \int_0^1 \psi_i' \psi_j' \, dx \qquad \text{stiffness matrix}$$

$$M_{ij} = \int_0^1 \psi_i \psi_j \, dx \qquad \text{mass matrix}$$

$$F_i = \int_0^1 \psi_i(x) f(x) \, dx$$

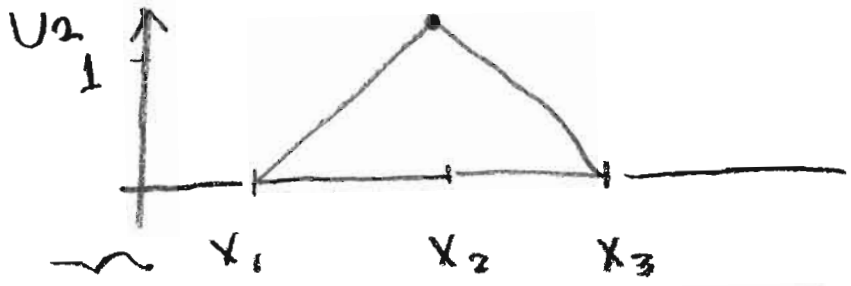
Almost always, the $\psi_j(x)$ functions are piecewise polynomials, so the K_{ij} and M_{ij} entries can be evaluated exactly using analytical formulae, or Gaussian quadrature of sufficiently high order. The F_i are approximated with quadrature methods (typically Gaussian quadrature)

Q: What is S and the basis functions $\{\psi_j\}$ that go with it?

Many choices, even in 1D... they all begin with an underlying choice of subintervals of $[0,1]$:

(I) $S = \{$ piecewise linear, continuous functions $\}$

dimension N, basis functions



zero outside adjacent intervals.

Note that in this case, the U_j coefficient value is an approximate value of $U(x)$, i.e. $U_j = U(x_j)$. This is not always the case.

Note: If we take $h := \max_j |x_j - x_{j-1}|$

↑ define $x_0 = 0$

then it can be proved that $\|U - u\|_\infty \leq Ch^2$ where C depends on f. Here, $U(x)$ and $u(x)$ are continuous functions of x, so we mean

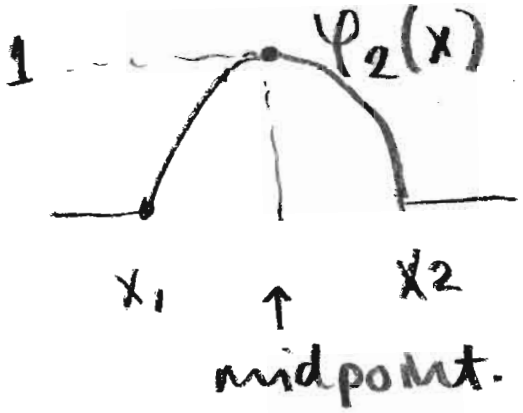
$$\|U - u\|_\infty = \max_x |U(x) - u(x)|.$$

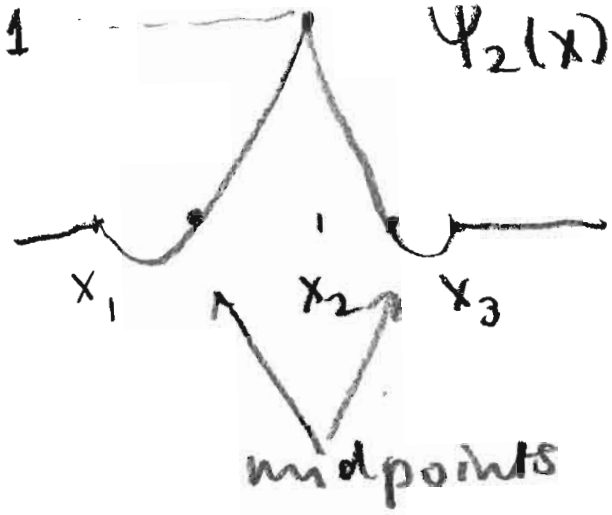
i.e. convergence not just at grid points

$O(h^3)$ convergence.

(II) $S = \{$ piecewise quadratic, continuous functions $\}$.

dimension $2N$, two types of basis functions.





Here again, function coefficients represent approximate values.

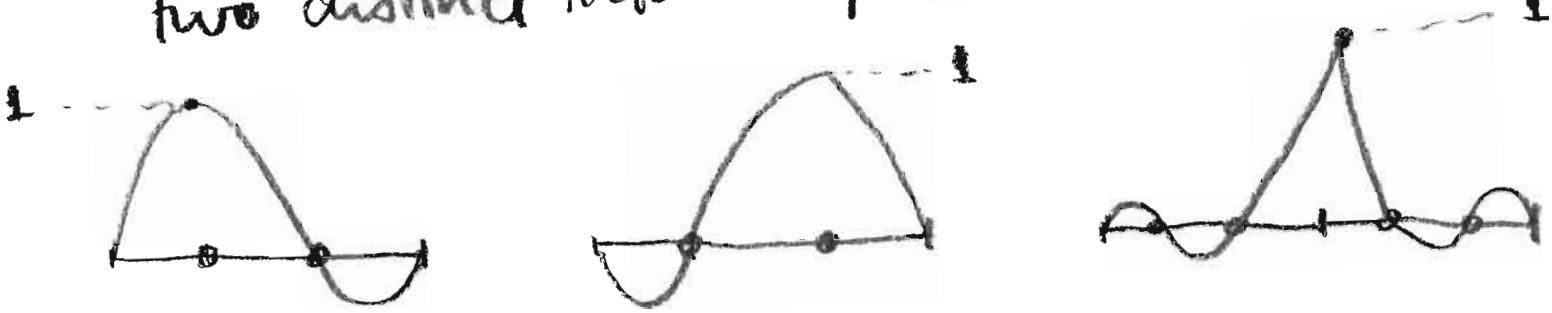
(III) $S = \{$ piecewise cubic, C_1 functions $\}$.

$O(h^4)$ convergence. Coefficients are function value and derivative values at subinterval ends. dimension $2N$.

Hey! We have come full circle! This is a Hermite polynomial approximation! Well, it is polynomial approximation on every subinterval that we get from an approximation of the weak form of the problem.

(IV) $S = \{$ piecewise cubic, continuous functions $\}$.

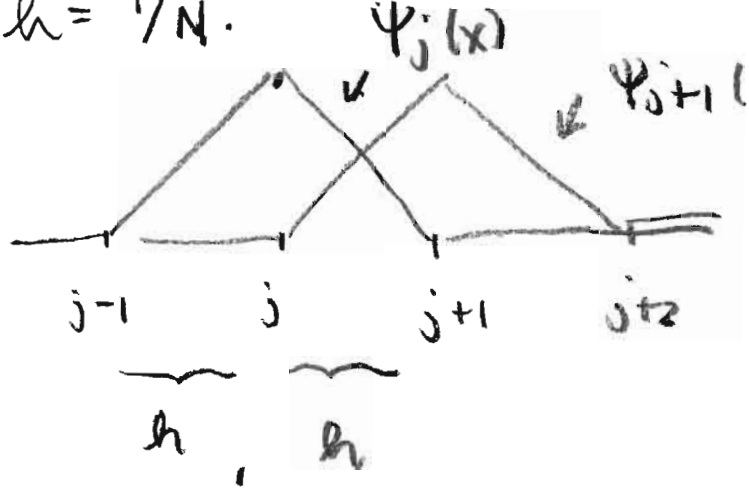
dimension $3N$, coefficients are the values at subinterval ends and two distinct interior points.



use ideas from interpolation & quadrature: subintervals are mapped to a reference interval ($[-1,1]$ or $[0,1]$), interpolation and quadrature are done on the reference interval.

Clearly, a computer code that did all of this work automatically from a given grid would be very useful. The process of generating the entries of the matrix that correspond to a given problem and a given grid and element choice is called matrix assembly.

As an example, let's calculate K & M for piecewise linear elements on a regular grid with N subintervals, length $h = 1/N$.



$$M_{ij} = \int_0^1 \psi_i \psi_j \, dx = 0 \text{ unless } j = i, i+1, i-1$$

$$M_{ii} = 2\int_0^h \left(1 - \frac{x}{h}\right)^2 dx = \begin{array}{l} \text{subs. } y = x/h \\ dx = h\,dy \end{array}$$

$$= 2h\int_0^1 (1-y)^2 dy = -\frac{2h}{3}(1-y)^3\Big|_0^1 = \frac{2h}{3}.$$

↑ reference interval

$$M_{ij} \; (j = i \pm 1) = \int_0^h \left(1 - \frac{x}{h}\right)\left(\frac{x}{h}\right) dx = h\int_0^1 (1-y)y\,dy = h\left(\frac{y^2}{2} - \frac{y^3}{3}\right)\Big|_0^1$$

$$= h/6.$$

$$K_{ij} = \int_0^1 \psi_i' \psi_j' \, dx = 0 \text{ unless } j = i, i+1, i-1.$$

$$K_{ii} = 2/h, \quad K_{i,i+1} = K_{i,i-1} = -\frac{1}{h}.$$

It can be shown that

$$F_i = \int_0^1 f(x)\,\psi_i(x)\,dx = h\,f(ih) + O(h^3).$$

So the FEM in this case $(M+K)\underline{U} = E$ reads for each equation as

$$-\frac{1}{h}U_{i-1} + \frac{2}{h}U_i - \frac{1}{h}U_{i+1} + \left(\frac{h}{6}U_{i-1} + \frac{2h}{3}U_i + \frac{h}{6}U_{i+1}\right) = F_i \qquad (4)$$

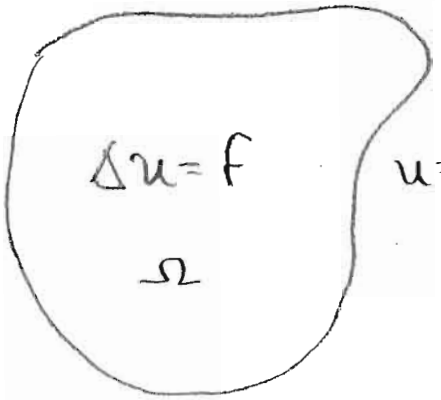
Dividing by h, and recognizing that

$$\frac{1}{6}U_{i-1} + \frac{2}{3}U_i + \frac{1}{6}U_{i+1} = u(ih) + O(h^2)$$

we see that (4) is similar to a FD discretization.

But: The FEM approach can easily handle non-uniform grids in 1D and complex domains in higher dimensions.

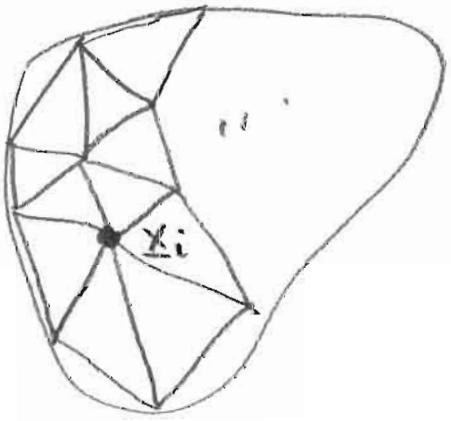
Math 405/607, Fall 2014, Day 21



Weak formulation - multiply by a test function $\varphi(x)$ and integrate by parts ($\varphi \equiv 0$ on $\partial\Omega$).

$$-\int_\Omega \underbrace{(u_x \varphi_x + u_y \varphi_y)}_{(\nabla u)\cdot(\nabla \varphi)} dA = \int_\Omega f \varphi \, dA \qquad (1)$$

Triangulate the domain (heuristic algorithms)



Note: can have an approximation of the domain with quadrilaterals, and the elements can be deformed at the boundary to better match the shape.

Let M be the number of interior nodes. Let $\psi_i(x,y)$ be the piecewise linear function on each triangle, with $\psi_i(x_i) = 1$ and $\psi_i(x_j) = 0$ $j \neq i$

↑ other nodes

Now $U(\underline{x}) = \sum_{i=1}^{M} U_i \Psi_i(\underline{x})$

is a continuous, piecewise linear (graph is a plane) approximation of $u(x)$, when it satisfies (1) for every $\Psi(\underline{x}) = \Psi_i(\underline{x})$.

Collect the M values of U_i in a vector $\underline{U}$. The resulting system is

$$-K\underline{U} = \underline{F}$$

where $K_{ij} = \int_{\Omega} \nabla\Psi_i \cdot \nabla\Psi_j \, dA$ (sparse).

and $F_i = \int_{\Omega} f(\underline{x}) \Psi_i(\underline{x}) \, dA.$

The integrals above are done on a reference element (the unit equilateral or unit right angle triangle).

Thus: quadrature methods on reference elements in 2D & 3D are needed.

The process of constructing (assembling) the entries of K_{ij} & F_i by integrating over triangles can be automated by a computer code library.

I will show you the "pdetool" in MATLAB.

Convergence analysis of FEM approximations involves some new function spaces and can get a bit technical.

Both analysis and implementation of FEM are covered in more detail in Math 521 next term.

More numerical linear algebra and an introduction to numerical optimization in CS 406, also next term.

Introduction to Scientific Computation: Time Stepping Methods for Initial Value Problems

Brian Wetton *

October 9, 2014

1 Initial Value Problems

We will consider time stepping schemes for general ODE initial value problems for $u(t)$:

$$\frac{du}{dt} = f(u(t), t) \ , \quad u(0) = u_0 \text{ given.} \tag{1}$$

We will consider this general form for scalar u when we assess the accuracy of time-stepping schemes, but all the schemes considered will be applicable to the vector $\mathbf{u}$ case. It is the general form since, as discussed last lecture, higher order DEs and higher order systems can always be converted to first order systems. We will also consider the simple scalar problem

$$\frac{du}{dt} = \lambda u \ , \quad u(0) = 1. \tag{2}$$

with λ a given complex constant to investigate the stability of time stepping schemes.

We will often use Newton's notation for time derivatives:

$$\frac{du}{dt} := \dot{u}, \quad \frac{d^2u}{dt^2} := \ddot{u}, \text{ etc.}$$

2 Basic Time Stepping Schemes and Ideas

Consider approximating $u(t)$ on a regular grid in time with interval size $k = \Delta t$. We will use superscripts for the time level index (since later we will look at PDE problems where we will discretize in space and time).

$$U^n \approx u(nk).$$

* wetton@math.ubc.ca

The simplest method for approximating solutions to (1) is the Explicit Euler (Forward Euler) method:

$$U^{n+1} = U^n + kf(U^n, nk). \tag{3}$$

We call this scheme a *one-step method* since the value of the approximation at time level n determines the values at time level $n+1$. This is a convergent scheme with first order convergence: If the solution of (1) is well defined in $[0, T]$ and (3) is used with N time steps of size $k = T/N$ then

$$|u(T) - U^N| = O(k).$$

Note that if the exact solution is put into (3) we get an expression for the truncation error from linear Taylor approximation, since $f(u(nk), nk) = \dot{u}(nk)$:

$$u((n+1)k) - u(nk) - kf(u(nk), nk) = \frac{k^2}{2}\ddot{u}(\xi)$$

for some $\xi \in (nk, (n+1)k)$. Thus, the local error (after one time step) is $O(k^2)$. It makes sense that the error after $O(1/k)$ times steps is first order, $O(k)$. The local error is usually written as $k\tau$, where τ is the truncation error. For the Forward Euler method,

$$\tau = \frac{k}{2}\ddot{u}(\xi). \tag{4}$$

Theorem 1 (ODE Theory) *If $f(u,t) \in C_n$ with $n \geq 1$, then the solution $u(t)$ of (1) exists, is unique and in C_{n+1} in a neighbourhood of $t = 0$.*

Theorem 2 (Dahlquist) *Any "reasonable" one-step time stepping method converges to the exact solution at times for which it is defined with an order of convergence equal to the order of the truncation error.*

Here, "reasonable" means that the scheme is consistent and has a very basic stability property. Since all of the schemes we will discuss below are "reasonable", this theorem does not help us choose a scheme suitable for a specific problem.

Consider the FE scheme (3) applied to the scalar problem (2). The exact solution is

$$u(t) = e^{\lambda t}$$

and the discrete solution satisfies $U_0 = 1$, $U^{n+1} = (1 + k\lambda)U^n$, so

$$U^n = (1+z)^n \text{ , with } z = k\lambda.$$

It is clear that the so-called *Growth Factor* $G(z) = 1 + z$ for the scheme should approximate e^z when z is small, as it does.

We consider the set of complex z such that $|G(z)| \leq 1$. This is known as the *stability region* of the method. For FE,

$$|G(z)| = |1 + z| \leq 1$$

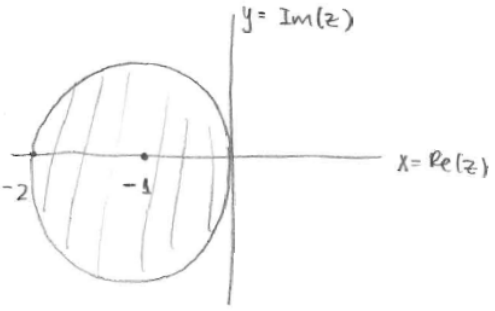


Figure 1: Stability region for Forward Euler time stepping.

is a circle of radius 1 in the complex plane, centred at $z = -1$ as shown in Figure 1. Note that if λ is real and negative, the exact solution decays in time, but the FE approximation will only decay if

$$|z| = |k\lambda| < 2 \text{ , that is } k < 2/\lambda. \tag{5}$$

This does not violate Dahlquist's theorem above, since as $k \to 0$, (5) is eventually satisfied.

3 Higher order and implicit methods

A second order explicit method called Improved Euler is given below:

$$\begin{aligned} U^* &= U^n + kf(U^n, nk) \\ U^{n+1} &= U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^*, (n+1)k)\right). \end{aligned}$$

This is a *two-stage* method, with two evaluations of the right hand side function f and U^* is the value at the intermediate stage. However, since U^n does determine U^{n+1} it is still a *one-step* method. It is one of a family of second order Runge-Kutta methods. It can be written more compactly:

$$U^{n+1} = U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^n + kf(U^n, nk), (n+1)k)\right). \tag{6}$$

From this form, the truncation error can be identified:

$$\tau = \frac{1}{k}\left[u^{n+1} - u^n - \frac{k}{2}(f^n + f(u^n + kf^n, (n+1)k))\right]$$

where $u^n := u(nk)$ and $f^n := f(u^n, nk) = \dot{u}(nk)$. We can expand all variables in Taylor series at $t = nk$, to obtain

$$\tau = \frac{1}{k}\left[k\dot{u} + \frac{k^2}{2}\ddot{u} + \frac{k^3}{6}\frac{d^3u}{dt^3} - \frac{k}{2}\dot{u} - \frac{k}{2}\left(\dot{u} + k(f_u f + f_t) + \frac{k^2}{2}(f_{uu}f^2 + f_{tt} + f_{ut}f)\right) + O(k^4)\right]$$

where all functions are evaluated at $t = nk$ and $u = u(nk)$. Starting with $\dot{u} = f(u,t)$ it can be shown that

$$\ddot{u} = f_u f + f_t$$

and

$$\frac{d^3u}{dt^3} = f_{uu}f^2 + 2f_{ut}f + (f_u)^2 f + f_u f_t + f_{tt}.$$

Using the first of the results above, we see that τ is second order

$$\tau = k^2\left(\frac{1}{6}\frac{d^3u}{dt^3} - \frac{1}{4}(f_{uu}f^2 + f_{tt} + f_{ut}f)\right) + O(k^3).$$

Note that the dominant error term in the truncation error is not a simple time derivative of u as it was for the FE method. This has implications for error estimation in adaptive methods as we shall see in later discussion.

We can consider the stability of the Improved Euler scheme by considering the form (6) with $f(u,t) = \lambda u$, giving

$$U^{n+1} = (1 + z + \frac{z^2}{2})U^n$$

which defines a growth factor $G(z) = 1 + z + z^2/2$. The stability region, $\{z : |G(z)| \leq 1\}$ is shown in Figure 2. As with FE, it is seen that IE is not suitable for problems that have λ with a large negative real part. Such problems are called *stiff* problems. We give a more complete definition below.

Definition 1 (Stiff ODE systems) *Solutions of a well-behaved ODE system can be multiplied by $e^{-\beta t}$ with some $\beta > 0$ so that they are bounded. The value of β can always be chosen and time scaled so that the resulting transformed system, when linearized around desired solutions, has $O(1)$ eigenvalues (components that evolve on an $O(1)$ time scale). A system is called* stiff *if the linearization of this transformed system also has eigenvalues of large size.*

Theorem 3 (Dahlquist) *The stability region of every explicit scheme is bounded.*

From this, it can be seen that no explicit scheme is suitable for stiff problems.

There are two properties we would like to have for a time-stepping scheme for stiff problems, summarized in the following definitions.

Definition 2 (L-stability) *A time-stepping scheme is called L-stable if $|G(z)| < 1$ for all z with $\Re(z) < 0$. That is, the stability region contains the left half plane.*

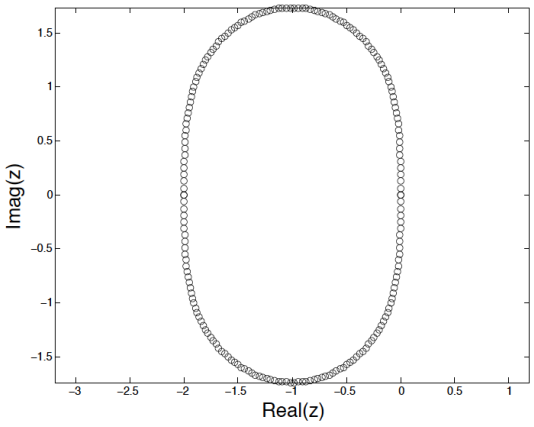


Figure 2: Stability region for Improved Euler time stepping (inside the curve shown). This is computed by setting $G(z) = e^{i\theta}$ for a grid of θ points and plotting the roots of the quadratic.

Definition 3 (A-stability) *A time-stepping scheme is called A-stable if*

$$G(z) \to 0 \quad as \quad \Re(z) \to -\infty.$$

This matches the property that $G(z)$ should approximate e^z.

Important Note: In some literature, the "A" and "L" definitions are reversed! There were two groups that could not agree on notation. I like "L" for left half-plane and "A" for asymptotic.

From Theorem 3 we see that no explicit scheme is L-stable or A-stable. Thus we turn to implicit schemes. The simplest scheme of this type is the Backward Euler scheme:

$$U^{n+1} = U^n + kf(U^{n+1}, (n+1)k).$$

It is called an implicit method because U^{n+1} is specified by an implicit relationship above. A linear or nonlinear system must be solved for U^{n+1} at every time step. It is a first order method with truncation error

$$\tau = -\frac{k}{2}\ddot{u}(\xi) \tag{7}$$

for some $\xi \in (nk, (n+1)k)$. It has growth factor

$$G(z) = \frac{1}{1-z}$$

and so has a stability region that is outside the unit circle centred at $z = 1$ as shown in Figure 3. From the form of $G(z)$ above and the shape of the stability region, it is clear that BE is both L-stable and A-stable. Thus, it is suitable for application to stiff problems.

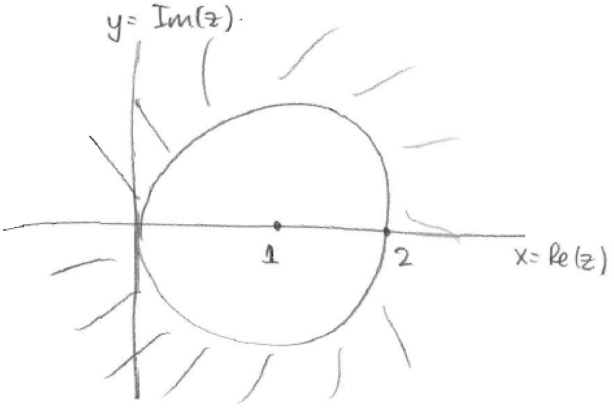


Figure 3: Stability region for Backward Euler time stepping.

4 Higher Order Implicit Schemes

We consider four higher order implicit schemes, each with some advantages and some disadvantages.

4.1 Trapezoidal Rule

We consider the Trapezoidal Rule

$$U^{n+1} = U^n + \frac{k}{2}\left(f(U^n, nk) + f(U^{n+1}, (n+1)k)\right)$$

and the Implicit Midpoint Rule

$$U^{n+1} = U^n + kf((U^{n+1} + U^n)/2, (n+1/2)k).$$

These are different schemes but become identical when applied to constant coefficient, linear, autonomous problems. They have the same dominant error term and the same growth factor and stability regions. When the Trapezoidal Rule is applied to diffusion problems, it is also known as the Crank-Nicholson method, and this name is sometimes used for the approach applied to other problems.

The growth factor is

$$G(z) = \frac{1 + z/2}{1 - z/2}$$

Here, the stability region is *exactly* the left half plane, and the method is thus L-stable. However,

$$G(z) \to -1 \quad \text{as} \quad \Re(z) \to -\infty.$$

so the method is not A-stable. Care should be used in the application of the method to stiff problems since components that should almost decay to zero in one time step instead just oscillate in sign. On the other hand, it does have desirable properties: L-stable, second order with a small error constant, one-step and one-stage.

4.2 Second Order Backward Differentiation Formula (BDF2)

This is a multi-step method, involving the values of two previous time steps, U^n and U^{n-1}. The first step, U^1, can be computed with BE without loss of overall accuracy.

$$U^{n+1} = \frac{4}{3}U^n - \frac{1}{3}U^{n-1} + \frac{2k}{3}f(U^{n+1}, (n+1)k)$$

It is based on the second order, one sided difference formula we derived earlier in the course

$$\dot{u}((n+1)k) \approx \frac{3U^{n+1} - 4U^n + U^{n-1}}{2k}.$$

It has truncation error

$$\tau = \frac{2}{9}k^2\frac{d^3u}{dt^3}(\xi)$$

with $\xi \in ((n-1)k, (n+1)k)$. To analyze the stability, consider the method applied to (2):

$$U^{n+1} = \frac{4}{3}U^n - \frac{1}{3}U^{n-1} + \frac{2}{3}zU^{n+1}$$

or

$$(1 - \frac{2z}{3})U^{n+1} - \frac{4}{3}U^n + \frac{1}{3}U^{n-1} = 0.$$

For a given z, this is a second order constant coefficient difference equation with solution

$$U^n = AG_1^n + BG_2^n \tag{8}$$

for some constants A and B and $G_1(z)$, $G_2(z)$ roots of

$$(1 - \frac{2z}{3})G^2 - \frac{4}{3}G + \frac{1}{3} = 0.$$

It is important to clarify that the superscript n for U in (8) is the time level, but for G_1 and G_2 it is an exponent. The stability region for two-step schemes is

$$\{z : |G_1(z)| < 1 \text{ and } \quad |G_2(z)| < 1\}$$

The stability region for BDF-2 is shown in Figure 4. It is clear that it is L-stable. Since

$$G_{1,2} = \frac{\frac{2}{3} \pm \sqrt{\frac{4}{9} - \frac{1}{3}(1 - 2z/3)}}{1 - 2z/3} \tag{9}$$

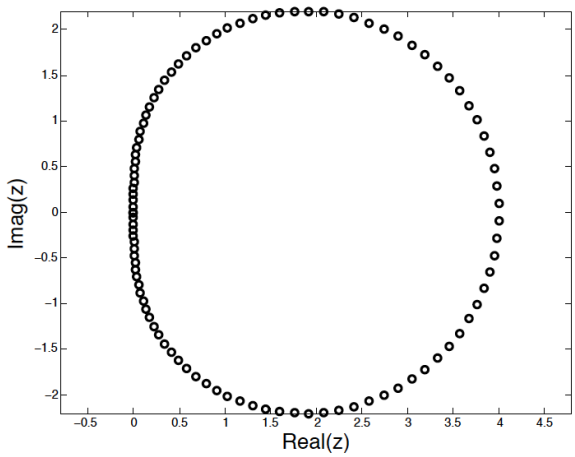


Figure 4: Stability region for BDF-2 (outside the curve shown). This is computed by setting $G(z) = e^{i\theta}$ for a grid of θ points and plotting $z = 3/2 - (2G - 1/2)/G^2$.

it is also clear that the scheme is A-stable ($|G_{1,2}| = O(|z|^{-1/2})$ in the limit $|z| \to \infty$). Considering (9) in the limit as $z \to 0$ (think of fixed λ, $k \to 0$), we have

$$G_1 \approx 1 \ , \quad G_2 \approx \frac{1}{3}.$$

The term G_2^n appears as an initial layer that accounts for the error from the initialization procedure for U^1.

4.3 Second Order Diagonally Implicit Runge-Kutta (DIRK2)

This is a one-step, two-stage implicit method:

$$\begin{aligned} U^* &= U^n + k\alpha f(U^*, (n+\alpha)k) \\ U^{n+1} &= U^n + k\left(\alpha f(U^{n+1}, (n+1)k) + (1-\alpha) f(U^*, (n+\alpha)k)\right). \end{aligned}$$

with $\alpha = 1 - \sqrt{2}/2$. The term "diagonal" applies since each stage is only implicit in the value at that stage. It is both A-stable and L-stable. Since it is a one-step method, adaptive methods for time step size are more easily implemented than for BDF-2. Over one time step, it is more accurate than BDF-2 under certain assumptions (the error terms are not directly comparable in general). However, taking into account the fact that BDF-2 has one implicit solve per time step and DIRK-2 has two, BDF-2 is more efficient for fixed time step computations.

4.4 Radau II-A

This is a third order, two stage method that is both L-stable and A-stable.

$$\begin{aligned} U^* &= U^n + k\left(\frac{5}{12}f(U^*,(n+1/3)k) - \frac{1}{12}f(U^{n+1},(n+1)k)\right) \\ U^{n+1} &= U^n + k\left(\frac{3}{4}f(U^*,(n+1/3)k) + \frac{1}{4}f(U^{n+1},(n+1)k)\right). \end{aligned}$$

Note that this scheme is implicit in both U^* and U^{n+1} simultaneously.

5 Adaptive Time Stepping

We discuss adaptive time stepping with a particular, simple example. Consider applying BE to a problem and wanting to make the local error at each time step smaller than a user defined tolerance δ. The local error is (7):

$$k\tau = -\frac{k^2}{2}\ddot{u}(\xi_1) \approx -\frac{k^2}{2}\ddot{u}(nk)$$

The local error for FE (4) is

$$k\tau = \frac{k^2}{2}\ddot{u}(\xi_2) \approx \frac{k^2}{2}\ddot{u}(nk)$$

So, if we compute the FE solution U^{FE} as well as the BE solution U^{n+1}, the local error E in U^{n+1} is approximately

$$E \approx \frac{1}{2}|U^{n+1} - U^{FE}|.$$

Note that U^{FE} is cheap to compute (an explicit method) and may be useful as a good initial guess for iterative solvers for U^{n+1}. The value of U^{FE} is not used in subsequent calculations, so there is no stability problem.

From the value of E computed above, we would make some decisions about the step from t_n to $t_{n+1} = t_n + k$:

If $E > \delta$: We would fail the time step and recompute U^{n+1} with a reduced time step k.

If $E \leq \delta$: We would accept the time step and would compute a new time step k so that $E \leq \delta$ would be likely to be satisfied again. Since

$$E \approx Ck_{old}^2$$

(with $C \approx |\ddot{u}(t_n)|/2$) and we want $E \leq \delta$ we could take k_{new} with

$$Ck_{new}^2 \approx \frac{E}{k_{old}^2}k_{new}^2 \leq \delta.$$

In practice, the formula

$$k_{new} = \theta k_{old}\sqrt{\delta/E}$$

is used, with $\theta < 1$ a computational parameter, a "safety" factor. I typically use $\theta = 0.8$. There is also typically a limit on how much k is allowed to increase. On a failed step for which E is not that much bigger than δ, the formula above can also be used. On a "bad" failure, k is typically reduced drastically (i.e. by a factor of two).

Note that higher order single step methods have more complicated error structure. For these methods, typically higher order methods are used to assess accuracy of lower order methods. For example, MATLAB's `ode45` explicit solver uses a fifth order Runge-Kutta method to assess the accuracy of a fourth order one. It uses the Dormand-Prince pair, which is a six stage method carefully chosen so that the same function evaluations are used for both methods.

6 Butcher Tables

The coefficients of one-step methods can be represented in Butcher Tables. The general s stage method (s function evaluations) is given by

$$\begin{aligned} U^{n+1} &= U^n + \sum_{i=1}^{s} b_i K_i \\ K_i &= kf\left(U^n + \sum_{j=1}^{s} a_{ij} K_j, t_n + c_i k\right) \end{aligned}$$

The constants in the general method above can be entered into a table:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

An explicit method has zeros in the diagonal of $\mathbf{A}$ and above. A diagonally implicit method has non-zeros on the diagonal but zeros above. The tables for some of the schemes discussed in this section are given below:

Improved Euler:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

DIRK-2: with $\alpha = 1 - \sqrt{2}/2$

α	α	0
1	$1-\alpha$	α
	$1-\alpha$	α

Radau IIA:

$1/3$	$5/12$	$-1/12$
1	$3/4$	$1/4$
	$3/4$	$1/4$

Spectral Methods

We consider our 1D elliptic model problem again:

$$Au := -u_{xx} + u = f \tag{1}$$

with $x \in [0, 1]$ and u periodic. The continuous fourier transform for 1-periodic functions is given by

$$\hat{u}_\alpha = \mathcal{F}(u) = \int_0^1 u(x)\, e^{-2\pi i \alpha x}\, dx \tag{2}$$

with inverse

$$u(x) = \sum_{\alpha=-\infty}^{\infty} \hat{u}_\alpha e^{2\pi i \alpha x}. \tag{3}$$

Note that $\{e^{2\pi i \alpha x}\}$ is an orthonormal set in L_2 norm and spans L_2 and

$$\|u\|^2 = \sum_{\alpha=-\infty}^{\infty} |\hat{u}_\alpha|^2.$$

Also, these functions are eigenfunctions of A, **i.e.**

$$A\, e^{2\pi i \alpha x} = (4\pi^2\alpha^2 + 1) e^{2\pi i \alpha x}.$$

That A has a complete set of orthogonal functions is not such a surprise - since A is symmetric and positive definite (A is unbounded which makes the proof of this a little more difficult).

An expression for the solution to (1) can be found easily by decomposing f into spectral representation, **i.e.** we can find $\mathcal{F}(f)$, then $\hat{u}_\alpha = \hat{f}_\alpha/(4\pi^2\alpha^2 + 1)$ and $u(x)$ can be found by the inverse transform.

spectral methods

The idea of spectral methods is to use a finite number of terms in the expansion (3). We will use an approximation

$$U(x) = \sum_{-N/2}^{N/2} \hat{u}_\alpha e^{2\pi i \alpha x}.$$

where we use

$$\hat{u}_\alpha = \frac{\hat{f}_\alpha}{4\pi^2\alpha^2 + 1}$$

with the exact $\hat{f}_\alpha$ for now. This is a functional approximation just like the FE one. We can ask how close U is to the exact u in some convenient norm. For this problem, all norms are convenient. We'll use the L_2 norm here and notice that

$$\|U - u\|^2 = \sum_{|\alpha|>N/2} |\hat{u}_\alpha|^2 = \sum_{|\alpha|>N/2} |\hat{f}_\alpha|^2/(4\pi^2\alpha^2 + 1)^2 \tag{4}$$

since the coefficients for small α are the same. We assume that $f \in C^\infty$ (periodic) and derive the following simple estimate:

$$\begin{aligned} |\hat{f}_\alpha| &= |\int_0^1 f(x) e^{-2\pi\alpha x}\, dx| \\ &= |\int_0^1 \frac{f'(x)}{-2\pi i\alpha} e^{-2\pi\alpha x}\, dx| \\ &\leq B/|\alpha| \end{aligned}$$

where $B = \max|f'|/(2\pi)$. We repeat integration by parts to get

$$|\hat{f}_\alpha| \leq B_j/|\alpha|^j \tag{5}$$

where $B_j = \max|f^{(j)}/(2\pi)^p$. This shows that if f is smooth, then $|\hat{f}_\alpha|$ decays rapidly with $|\alpha|$. Returning to the error equation (4) we see that

$$\begin{aligned} \|U-u\|^2 &= \sum_{|\alpha|>N/2} |\hat{f}_\alpha|^2/(4\pi^2\alpha^2+1)^2 \\ &\leq \left(\frac{B_j}{(N/2)^j}\right)^2 \sum_{|\alpha|>N/2} \frac{1}{(4\pi^2\alpha^2+1)^2} \\ &\leq (C_j/N^j)^2 \end{aligned}$$

for all j where C_j depends only on the size of the derivatives of f up to order j (note that this estimate is not sharp). Thus we have

$$\|U-u\| \leq C_j/N^j \tag{6}$$

for all j. Recall the estimate for a second order FE approximation:

$$\|U_h - u\| \leq Ch^2.$$

For higher order (q) methods, we will see higher order convergence

$$\|U_h - u\| \leq Ch^q$$

but q is still fixed. Considering (6) we see that spectral methods are asymptotically more accurate that any FE or FD method since they converge faster than any power of $h = 1/N$. This type of convergence (6) is called spectral convergence. Remember that this estimate only holds for smooth data f.

Aliasing and Pseudo-Spectral Methods

The only thing we don't like about the spectral method described above is the potentially laborious calculation to accurately compute $\hat{f}_\alpha$ from the integrals (2) and the evaluation of $U(x)$ at desired points from the summation (3) (recall this summation has only finite terms for U). The way to get around this problem is to use the

inverse FFT to evaluate the approximation on a uniform grid with N grid points - this procedure is fast and exact. We can also use the FFT to approximate the fourier coefficients $\hat{f}_\alpha$ from a vector of values $F_i = f(ih)$ on N grid points. The details are presented below.

Recall that the DFT is given by

$$\hat{U}_\alpha = \frac{1}{N} \sum_{k=0}^{N-1} U_k e^{-2\pi i k\alpha/N}.$$

and the inverse is given by

$$U_k = \sum_{\alpha=0}^{N-1} \hat{U}_\alpha e^{2\pi i k\alpha/N}. \tag{7}$$

Note that different scalings from the original scalings have been used here. We will first deal with some technical details. Note that the sum above goes from 0 to $N-1$ but the spectral method provides us with $\hat{u}_\alpha$ for $|\alpha| \leq N/2$ (this makes more sense because the smaller $|\alpha|$ are the significant ones). However, if we evaluate the functions $e^{2\pi i\alpha x}$ and $e^{2\pi i(\alpha+N)x}$ only at the points $x_k = k/N$ on the grid, we cannot distinguish them. Therefore, we can easily relabel the summation in (7) to go from $\alpha = -N/2+1, \ldots N/2$ where

$$\hat{U}_\alpha \sim \hat{u}_\alpha \quad \text{for } \alpha = -N/2+1, \ldots N/2-1$$

and

$$\hat{U}_{N/2} \sim \hat{u}_{-N/2} + \hat{u}_{N/2}. \tag{8}$$

No information has been lost in (8) because these two modes are indistinguishable on the grid. Whatever we do with the $N/2$ coefficient is not that important since we expect it to be very small (remember the fast decay for large α). However, it is sometimes useful to consider this term in the form (8), **i.e.** so that spectral evaluations of first derivatives of real f stay real. Using the above values of $\hat{U}$, the inverse FFT can be used to evaluate the function U on the grid points exactly.

To approximate $\hat{f}_\alpha$ for $|\alpha| \leq N/2$ we follow the same plan. We get the vector F by evaluating f on the grid points and then compute $\hat{F}$ by the inverse FFT and identify

$$\hat{F}_\alpha \sim \hat{f}_\alpha \quad \text{for } \alpha = -N/2+1, \ldots N/2-1$$

and

$$\hat{F}_{N/2} \sim \hat{f}_{-N/2} + \hat{f}_{N/2}. \tag{9}$$

It is easy to show that in fact

$$\hat{F}_\alpha = \sum_{l=-\infty}^{\infty} \hat{f}_{\alpha+Nl}. \tag{10}$$

This is not surprising since we cannot distinguish between the α, $\alpha+N$, $\alpha+2N$, etc. modes on the grid. The effect in (10), where high frequency information is seen as

low frequency information, is called aliasing. Using the decay of the coefficients (5) it can be shown that

$$|\hat{F}_\alpha - \hat{f}_\alpha| \leq C_j/N^j \tag{11}$$

for $|\alpha| \leq N/2 - 1$ and all j. A similar bound can be made for the $N/2$ term. With the $\hat{F}$ values we can compute

$$\hat{U}_\alpha = \hat{F}_\alpha/\kappa_\alpha$$

for $\alpha = -N/2 + 1, \ldots N/2$ where κ_α is the corresponding eigenvalue $4\pi^2\alpha^2 + 1$. Note that this approximation is consistent with (8) and (9) since $\kappa_{-N/2} = \kappa_{N/2}$. Now, the values of U can be evaluated on the grid as described above. This is a fast ($O(N \log N)$) method called the pseudo-spectral method which is also spectrally accurate as shown below.

We consider pointwise errors rather than L_2 errors this time.

$$\begin{aligned} |u(x) - U(x)| &= |\sum_{\alpha=-\infty}^{\infty} \frac{\hat{f}_\alpha}{\kappa_\alpha} - \sum_{\alpha=-N/2}^{N/2} \frac{\hat{F}_\alpha}{\kappa_\alpha}| \\ &\leq \sum_{\alpha=-N/2}^{N/2} |\hat{f}_\alpha - \hat{F}_\alpha| + \sum_{|\alpha|>N/2} |\hat{f}_\alpha|. \end{aligned}$$

The second term decays spectrally as before and so does the first using (11). Thus we have

$$|u(x) - U(x)| \leq C_j/N^j$$

for all j and all x. We write $\|u - U\|_\infty \leq C_j/N^j$. As before, the estimates above are not sharp.

Problems with Variable Coefficients

We now apply a spectral method to the 1D analogue of M3:

$$-u_{xx} + b(x)u = f$$

with $b(x) = (2+\cos(\cos(2\pi x))$ and u 1-periodic. This is a symmetric, positive definite problem which has a complete set of eigenfunctions as before. However, we do not want to use these functions as our basis because we don't know them and we won't have a fast transform technique to do the conversion from a grid representation to a spectral one. Therefore, we will continue to use the Fourier basis from before.

Note 1 (Terminology) *A spectral method does not mean we always use a spectral representation of the problem. It means we are using a representation that will give us spectral accuracy.*

We first develop infinite conditions that the exact solution u must satisfy. By taking the transform of the original problem we get

$$4\pi^2\alpha^2\hat{u}_\alpha + \widehat{bu}_\alpha = \hat{f}_\alpha.$$

However,

$$\widehat{bu}_\alpha = \sum_{n=-\infty}^{\infty} \hat{b}_n \hat{u}_{\alpha-n}$$

so the FT does not diagonalize the problem effectively (this should not be expected since the fourier terms are not eigenfunctions).

A true spectral method (a Galerkin method) can be derived by assuming $\hat{u}_\alpha = 0$ for $|\alpha| > N/2$ and then projecting the resulting equations on to the corresponding finite dimensional space (just like FE). The resulting equations for $\hat{U}_\alpha$ for $\alpha \leq N/2$ are

$$(-4\pi^2\alpha^2 I + B)\hat{U}_\alpha = \hat{f}_\alpha$$

where the matrix B is full (with the $\hat{b}$ terms from the truncated convolution). Solving this system directly will be slow, because full matrix methods must be used. Although solving the method iteratively by the CG method is possible (the matrix is symmetric and positive definite), it will be slow because multiplication by B will be slow.

We can speed up this process, however, by evaluating $B\hat{U}$ approximately (in the pseudo-spectral sense which will introduce aliasing). Recall that $B\hat{U}$ approximates $\widehat{bu}$. Therefore, we could compute U_i by inverse FFT, multiply *pointwise* by b_i and then compute the inverse transform. This approximation is equivalent to replacing the matrix B by

$$\mathcal{F}\tilde{B}\mathcal{F}^{-1}$$

where $\tilde{B}$ is a diagonal matrix representing the pointwise multiplication. The resulting system is

$$A_3\hat{U} := (\Xi + \mathcal{F}\tilde{B}\mathcal{F}^{-1})\hat{U} = \mathcal{F}F \tag{12}$$

where Ξ is a diagonal matrix with entries $4\pi^2\alpha^2$ and it is assumed that $\hat{f}$ is approximated pseudospectrally. Note that A_3 is a symmetric (since $\mathcal{F}^{-1} = \mathcal{F}^*$ where $*$ denotes complex transpose) and positive definite since $\tilde{B}$ and Ξ have positive entries. It is also possible to evaluate A_3 quickly so a CG method can be applied. In this case, preconditioning by the inverse of the pseudospectral approximation of the constant coefficient case described above should again prove effective.

Note 2 (Don't need A sparse for CG) *This example shows that a matrix does not have to be sparse to make CG an efficient technique. In fact, our matrix A_3 was dense, but we could still evaluate it quickly. For another interesting example of this phenomenon, see [3].*

Note 3 (Notation) *Since almost all practical methods involve pseudo-spectral evaluation of coefficients and interpolation, the "pseudo" is being dropped from these methods and they are usually called simply spectral methods.*

Be Careful with Boundaries

What about domains with boundaries? We consider a 1D analogue of M1

$$-u_{xx} = f$$

with $u(0) = u(1) = 0$. The eigenfunctions of this problem are $\{\sin(n\pi x)\}$ with corresponding eigenvalues $n^2\pi^2$. Since these correspond to the basis of the fast sine transform, it seems to be a suitable basis for a spectral method. We consider computing the problem with $f \equiv 1$ (smooth), giving $u(x) = -x^2/2 + x/2$ (also smooth). However, the coefficients in the sine series for f

$$\hat{f}_n = \sqrt{2} \int_0^1 f(x) \sin(n\pi x) dx$$

decay only like n^{-1}. The corresponding solution u has sine coefficients $\hat{u}$ that decay only like n^{-3}. Thus, the "spectral" approximation of this problem described above will have errors of size $h^3 = 1/N^3$, **i.e.** errors no better than a third order FD or FE method.

In this case, the eigenfunctions are a *poor* choice for the basis functions for a method. A more appropriate choice would be the Chebyshev polynomials (using fast interpolation on an irregular grid of points). The details are given in the references.

References

[1] Canuto, C. et. al., *Spectral Methods in Fluid Dynamics.*

[2] Gottlieb, D. and Orszag, S., *Numerical Analysis of Spectral Methods.*

[3] Rokhlin, V., "Rapid Solution of Integral Equations of Classical Potential Theory," JCP **60**, 187-207 (1983).

MATH 405/607E, Fall 2014, Wetton
Assignment #1 - due Thursday, September 25

Instructions: Do all questions in part A. Do one question in part B.

Part A Do all 8 questions. For part A questions, it is not necessary to be completely rigorous mathematically.

A1. We considered the matrix operator norm in lecture #1

$$\|\mathbf{A}\|_\infty := \max_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}$$

where the vector norm

$$\|\mathbf{x}\|_\infty := \max_{1\leq i\leq n} |x_i|$$

and n is the number of components of $\mathbf{x}$ with $\mathbf{A}$ $n \times n$. It can be shown that

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

(the maximum of the absolute row sums). Show that if

$$|b_{ij}| < C|a_{ij}|$$

for all i and j and a given constant $C > 0$, then

$$\|\mathbf{B}\|_\infty \leq C\|\mathbf{A}\|_\infty$$

A2. Find a formula involving the entries of a square matrix $\mathbf{A}$ for $\|\mathbf{A}\|_1$, the matrix operator norm corresponding to the vector norm

$$\|\mathbf{x}\|_1 := \sum_{i=1}^{n} |x_i|.$$

A3. Find the root of the following system near (1,1) ($x = 1$ and $y = 1$) with four decimal place accuracy. Use Newton's method and (1,1) as an initial guess. Indicate the number of iterations taken to reach the residual tolerance you specify. Include a print-out of your code.

$$\begin{aligned} f(x,y) &= x^2y^5 - x^5y^2 + e^{xy-1} \\ g(x,y) &= xy^3 + x^2y^2 + x^3y - 3 \end{aligned}$$

A4. Using equally spaced (distance h) function values $f(-2h)$, $f(-h)$, $f(0)$, $f(h)$, and $f(2h)$, derive:

(a) a second order accurate approximation for $f'''(0)$

(b) a fourth order approximation for $f''(0)$

A5. How many terms of the Taylor series for tanh based at $x = 0$ need to be taken to guarantee an accuracy of 0.001 for all $x \in [0, 1/2]$?

A6. Consider the Babylonian strategy to find square roots of numbers in the range [1,4]. Assume that the iterations start with $x_0 = 1$. Give the number of iterations necessary to give the result to an accuracy of 0.001 for all values in the interval.

A7. The function $\tanh x$ can be approximated by the bilinear function

$$B(x) = \frac{x}{1+x}.$$

Note that B matches the value and derivative of tanh at $x = 0$ and the behaviour as $x \to \infty$. Determine the maximum error in this approximation *for all* $x > 0$. Consider a more generalized approximation of the form

$$P(x) = \frac{C_0 + C_1 x + C_2 x^2}{1 + A_1 x + A_2 x^2}$$

where C_0, C_1, C_2, A_1, A_2 are constants. Determine values for these constants so that $P(x)$ has a smaller maximum error to $\tanh x$ for $x > 0$ than $B(x)$. Describe the reasoning for your choice of coefficients and how you determined them.

A8. Consider a cubic polynomial approximation based on function values and function second derivative values at the end-points of subintervals. Show that such an approximation is fourth order accurate.

A9. Derive a sixth order accurate Gaussian quadrature formula using function evaluations at three points.

Part B. Do one question. Be as rigorous as you can for these questions.

B1. Implement a spectral approximation for $e^{\cos x}$ on the interval $[0, 2\pi]$. Observe the interpolation errors to the value at $x = \sqrt{2}$ as $h \to 0$ ($N \to \infty$). Describe your approach but do not include code.

B2. Implement a function that returns the value of $J_0(x)$ (the Bessel function of first kind of order zero) accurate to 6 digits for $x \in [0, 10]$. Your function can have stored, pre-computed values, but otherwise can involve only arithmetic computations. Describe your approach but do not include code.

B3. Consider the matrix norm below

$$\|\mathbf{A}\|_* := \max_{i,j} |a_{ij}|.$$

Decide whether this is a matrix operator norm, that is whether or not there is a vector norm $\|\mathbf{x}\|_*$ such that

$$\|\mathbf{A}\|_* = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_*}{\|\mathbf{x}\|_*}.$$

If the result is true, describe the vector norm $\|\mathbf{x}\|_*$.

Part C. Challenge question. Not for marks, just "glory".

Consider the set of 10×10 invertible matrices whose entries are restricted to the integer values 0 to 9. Find the matrix in this set that has the largest condition number (based on the maximum norm). Prove your result.

Math 405/607 assignment #1 Solutions,
Part A

1. If $|b_{ij}| < C|a_{ij}|$ for all i,j then

$$\sum_{j=1}^{n} |b_{ij}| < C \sum_{j=1}^{n} |a_{ij}| \quad \text{for all } i \text{ and}$$

$$\max_i \sum_{j=1}^{n} |b_{ij}| < C \max_i \sum_{j=1}^{n} |a_{ij}|$$

thus, $\|B\|_\infty < C \|A\|_\infty$

2. Consider $M = \max_j \sum_{i=1}^{n} |a_{ij}|$, the maximum of the absolute column sums. Consider

$$\underline{y} = A\underline{x}.$$

$$y_i = \sum_j a_{ij} x_j$$

$$\|y\|_1 = \sum_i |y_i| \le \sum_i \sum_j |a_{ij}|\,|x_j| \qquad \text{exchange the order of summation on the right}$$

$$\le M \sum_{j=1}^{n} |x_j|$$

$$\le M \|\underline{x}\|_1.$$

Thus, $\|A\|_1 \le M$. Now consider the column j that maximizes the quantity M. Consider

$$A\underline{e}_j = \underline{A}_j, \qquad \|\underline{A}_j\|_1 = M.$$

($\underline{e}_j$: standard basis vector, $\|\underline{e}_j\|_1 = 1$; $\underline{A}_j$: j'th column of A)

Thus $\|A\|_1 = M$ ✓.

3. Code posted, v newton.m. Starting at (1,1) I got convergence to a residual tolerance (max norm) of 10^{-6} in 4 iterations. The resulting root approximation was

$$x \approx 1.714$$
$$y \approx 0.838$$

4. Both parts begin with Taylor series expansions based at the origin.

$$f(\pm h) = f(0) \pm h f'(0) + \frac{h^2}{2} f''(0) \pm \frac{h^3}{6} f'''(0) + \frac{h^4}{24} f''''(0) \pm \frac{h^5}{120} f^{(5)}(0) + \frac{h^6}{720} f^{(6)}(\xi)$$

$$f(\pm 2h) = f(0) \pm 2h f'(0) + 2h^2 f''(0) \pm \frac{4}{3} h^3 f'''(0) + \frac{2}{3} f''''(0) \pm \frac{4}{15} h^5 f^{(5)}(0) + \frac{4h^6}{45} f^{(6)}(\xi).$$

Consider approximations based on a linear combination of values:

$$\alpha f(-2h) + \beta f(-h) + \gamma f(0) + \varepsilon f(h) + \phi f(2h).$$

Put the Taylor series into this linear combination and match terms to get the desired approximation.

(a) $f(0)$: $\alpha + \beta + \gamma + \varepsilon + \phi = 0$ (i)

$f'(0)$: $-2\alpha - \beta + \varepsilon + 2\phi = 0$ (ii)

$f''(0)$: $2\alpha + \beta/2 + \varepsilon/2 + 2\phi = 0$ (iii)

$f'''(0)$: $-8\alpha - \beta + \varepsilon + 8\phi = 6/h^3$ (iv)

$f''''(0)$: $\frac{2}{3}\alpha + \frac{1}{24}\beta + \frac{1}{24}\varepsilon + \frac{2}{3}\phi = 0$ (v)

This is a solvable linear system for the coefficients. It can be solved numerically in MATLAB or symbolically in Maple (or other symbolic math package). By hand, (iii) & (v) $\Rightarrow$

$$\beta = -\varepsilon, \quad \alpha = -\phi, \text{ and } \quad \gamma = 0 \quad \text{(i)}$$

Then (ii) & (iv) read:

$$2\varepsilon + 4\phi = 0$$
$$2\varepsilon + 16\phi = 6/h^3.$$

so $\phi = \frac{1}{2h^3}$, $\varepsilon = -\frac{1}{h^3}$.

Note that the $f^{(5)}(0)$ term is then

$$\left(\frac{4}{15} + \frac{8}{3}\right) h^2$$

so the method,

$$f'''(0) \approx \frac{-f(-2h) + 2f(-h) - 2f(h) + f(2h)}{2h^3}$$

is second order accurate.

(b) Proceed as above but now

$$\alpha + \beta + \gamma + \varepsilon + \phi = 0$$
$$-2\alpha - \beta + \varepsilon + 2\phi = 0$$
$$2\alpha + \beta/2 + \varepsilon/2 + 2\phi = 1/h^2$$
$$-8\alpha - \beta + \varepsilon + 8\phi = 0$$
$$\tfrac{2}{3}\alpha + \tfrac{1}{24}\beta + \tfrac{1}{24}\varepsilon + \tfrac{2}{3}\phi = 0.$$
$$-4/15\,\alpha - \tfrac{1}{120}\beta + \tfrac{1}{120}\varepsilon + \tfrac{4}{15}\phi = 0.$$

Solution:
$\varepsilon = \beta = \frac{4}{3h^2}$
$\alpha = \phi = -\frac{1}{12h^2}$
$\gamma = \frac{5}{2h^2}$.

Note that the last equation is satisfied "for free" and so it can be seen that the resulting expression is fourth order accurate:

$$f''(0) \approx \frac{-f(-2h) + 16f(h) - 30f(0) + 16f(h) - f(2h)}{12h^2}$$

Alternate derivation

$$\frac{f(h) - 2f(0) + f(-h)}{h^2} = f''(0) + \frac{f''''(0)}{12}h^2 + O(h^4)$$

Use Richardson extrapolation,

$$f''(0) \approx \frac{4}{3}\left[\frac{f(h) - 2f(0) + f(-h)}{h^2}\right] - \frac{1}{3}\left[\frac{f(2h) - 2f(0) + f(-2h)}{(2h)^2}\right]$$

This gives the same approximation as above.

5. We could derive bounds on K_N, the maximum of the N'th derivatives of the function, and use the Taylor series error argument, but here in part A, let's investigate the difference numerically (comparing T_N to tanh on a fine grid).

$f(x) = \tanh x$ $\qquad \tanh(0) = 0$

$f'(x) = 1 - \tanh^2 x$ $\qquad f'(0) = 1.$

$f''(x) = -2\tanh x(1 - \tanh^2 x)$ $\qquad f''(0) = 0.$

$= -2\tanh x + 2\tanh^3 x$

$f'''(x) = (-2 + 6\tanh^2 x)(1 - \tanh^2 x)$ $\qquad f'''(0) = -2.$

$= -2 + 8\tanh^2 x - 6\tanh^4 x$

$f''''(x) = (16\tanh x - 24\tanh^3 x)(1 - \tanh^2 x)$ $\qquad f^{(4)}(0) = 0$

$= 16\tanh x - 40\tanh^3 x + 24\tanh^5 x$

$$f^{(5)}(x) = (16 - 120\tanh^2 x + 120\tanh^4 x)(1-\tanh^2 x)$$

$$f^{(5)}(0) = 16.$$

Thus, we can consider candidates (numerical approx.)

$T_3(x) = x - x^3/3$ $\quad \|f - T_3\|_\infty \approx 3.8e{-3}.$

$T_5(x) = x - x^3/3 + x^5/10$ $\quad \|f - T_5\|_\infty \approx 6.5e{-4}.$

Thus, the 5th order polynomial is sufficient.

Notes: By numerical approximation, I mean the MATLAB commands

```
x = linspace(0, 0.5, 1000);
max(abs(tanh(x) - x + x.^3/3 - x.^5/10)).
```

I then repeated with 10,000 points.

- Evaluating derivatives can be done with symbolic packages (Wolfram Alpha, Maple).
- Working out the k values analytically by substituting $y = \tanh x$, $y \in [0, \tanh 1/2]$.

6. To find $\sqrt{x}$ we would have the sequence.

$$x_0 = 1$$

$$x_1 = \tfrac{1}{2}(1 + x)$$

$$x_2 = \tfrac{1}{2}\left(x_1 + \frac{x}{x_1}\right) = \tfrac{1}{2}\left(\tfrac{1}{2}(1+x) + \frac{x}{\frac{1}{2}(1+x)}\right) = \cdots$$

$$= \frac{1+x}{4} + \frac{x}{1+x}$$

$$x_n = \tfrac{1}{2}\left(x_{n-1} + \frac{x}{x_{n-1}}\right).$$

as in the last question, I investigate this numerically

$$\|\sqrt{x} - x_1\|_\infty \approx 0.5$$

$$\|\sqrt{x} - x_2\|_\infty \approx 0.05$$

$$\|\sqrt{x} - x_3\|_\infty \approx 6.1e-4$$

So, 3 iterations are sufficient.

7. Again, investigate numerically

$$\|\tanh x - \frac{x}{1+x}\|_\infty \approx 0.3063$$

$\uparrow$

$x \geq 0$

Keep the same 3 features as the approximation above.

$$P(0) = 0 \Rightarrow C_0 = 0$$

$$P(\infty) = 1 \Rightarrow C_2 = A_2$$

$$P(x) = \frac{C_1 x + C_2 x^2}{1 + A_1 x + C_2 x^2}$$

$$P'(x) = \frac{C_1 + 2C_2 x}{1 + A_1 x + C_2 x^2} - \frac{(C_1 x + C_2 x^2)(A_1 + 2C_2 x)}{(1 + A_1 x + C_2 x^2)^2}$$

$P'(0) = C_1$, so $P'(0) = 1 \Rightarrow C_1 = 1$.

as an additional condition, match $P''(x) = 0$

$$P'(x) = \frac{(1 + 2C_2 x)(1 + A_1 x + C_2 x^2) - (x + C_2 x^2)(A_1 + 2C_2 x)}{(1 + A_1 x + C_2 x^2)^2}$$

$P''(0) = A_1 + 2C_2 - A_1 - 2A_1 = 2C_2 - 2A_1$

(can see this without computing $P''(x)\cdots$)

$P''(0) \Rightarrow A_1 = C_2 := C$

$$P(x) = \frac{x + Cx^2}{1 + Cx + Cx^2}$$

as a final condition, match

$P(1) = \tanh(1)$

$$\frac{1+C}{1+C+C} = \tanh(1)$$

$$1 + C = (1+2C)\tanh(1)$$

$$C = \frac{1 - \tanh(1)}{2\tanh(1) - 1}$$

With these values,

$\| \tanh x - P(x) \|_\infty \simeq 0.11833$

(Not a great improvement, you could do better).

8. Consider the approximation on the reference interval $[0,1]$. Let $C(x)$ be the cubic approx.

Consider $$q(x) = f(x) - C(x) - \frac{(x - 2x^3 + x^4)}{(b - 2b^3 + b^4)}(f(b) - C(b))$$

$q(0) = 0$, $q(1) = 0$ since $f(0) = C(0)$, $f(1) = C(1)$

and $(x - 2x^3 + x^4)\big|_{x=0,1} = 0$

Also note that $q(b) = 0$. assume $b \neq 0, 1$. Thus $q'(\xi_1) = q'(\xi_2) = 0$ with $0 < \xi_1 < \xi_2 < 0$. and thus $q''(\xi_3) = 0$ with $0 < \xi_3 < 1$.

Now note that $q'' = f'' - C'' - \dfrac{12x^2 - 12x}{(b - 2b^3 + b^4)}(f(b) - C(b))$

We see that $q''(0) = 0$ and $q''(1) = 0$ since $f''(0) = C''(0)$ and $f''(1) = C''(1)$ and $x^2 - x\big|_{0,1} = 0$.

Thus we have

$$q''(0) = 0, \quad q''(\xi_3) = 0, \quad q''(1) = 0$$

and we see there is a point $0 < \xi < 1$ so that

$$q^{(4)}(\xi) = 0$$

Since $C^{(4)} \equiv 0$ and $(x - 2x^3 + x^4)'''' = 24$ we

↑ cubic polynomial

have $f^{(4)}(\xi) - \dfrac{(f(b) - C(b))(24)}{b - 2b^3 + b^4} \Rightarrow$

$$f(b) - C(b) = \frac{f^{(4)}(\xi)}{24}(b - 2b^3 + b^4)$$

If we scale to an interval of length h, $f^{(4)}(\xi) = \dfrac{d^4 f}{dy^4}(\xi) \Rightarrow h^4 \dfrac{d^4 f}{dx^4}$, thus the approximation is fourth order accurate.

9. We must choose x_i and w_i, $i = 1, 2, 3$ so that

$$\left.\begin{aligned}
\sum w_i &= 2 \\
\sum x_i w_i &= 0 \\
\sum x_i^2 w_i &= 2/3 \\
\sum x_i^3 w_i &= 0 \\
\sum x_i^4 w_i &= 2/5 \\
\sum x_i^5 w_i &= 0.
\end{aligned}\right\}$$

6 nonlinear equations in 5 unknowns.

We could solve the first 5 using Newton's method, and see that the 6th is satisfied. With some insight, we can make easier progress.

Look for a solution with $x_1 = 0$, $x_{2,3} = \pm x$ and $w_{2,3} = w$. Thus, the even equations are all satisfied. Now

$$w_1 + 2w = 2$$

$$wx^2 = 1/3, \quad wx^4 = 1/5$$

These can be solved for $x^2 = 3/5$, $x = \sqrt{3/5}$, and $w = 5/9$, $w_1 = 8/9$.

Thus, $$\int_{-1}^{1} f(x)\,dx \approx \frac{5}{9} f\left(-\sqrt{3/5}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{3/5}\right)$$

MATH 405/607E, Fall 2014, Wetton
Assignment #2 - due Tuesday, October 14

Instructions: Do all questions in part A. Do one question in part B.

Part A Do all 5 questions. For part A questions, it is not necessary to be completely rigorous mathematically.

A1. Show that

$$\begin{aligned} D_+F_i &= \frac{F_{i+1}-F_i}{h} \\ D_-F_i &= \frac{F_i-F_{i-1}}{h} \end{aligned}$$

are first order accurate approximations of the first derivative. These are called forward and backward differences.

A2. (review) Find the exact solution of

$$-u''+u=x \quad \text{with } u(0)=0 \text{ and } u'(0)=-u.$$

Hint: Use the Method of Undetermined Coefficients.

A3. Modify the posted MATLAB code to approximate the boundary value problem in A2. Compare computed solutions to the exact solution and confirm second order convergence. Do not hand in code but describe how you implemented the right boundary condition and the details of your numerical convergence study.

A4. Consider the boundary value problem

$$-u''+u=f(x)$$

with u 2π-periodic in x. Consider approximating the problem with the fourth order accurate approximation of the second derivative from Assignment #1, question A4b. Analyze the stability of the scheme using Von Neumann Analysis.

A5. Consider the cubic approximation from Assignment #1, question A8, on the reference interval $y \in [0,1]$. Show that first derivatives approximated using the cubic polynomial at $y=0$ are third order accurate when the interval is mapped to one of length h. *Hint:* consider

$$q(y) = f(y) - C(y) - (f'(0) - C'(0))(y - 2y^3 + y^4)$$

Part B. Do one question. Be as rigorous as you can for these questions.

B1. Consider the following problem for $u(x)$, $x \in [0,1]$ where u can be discontinuous at $x = 1/2$. Using the MATLAB routine `bvp4c`, find the smallest non-zero value of λ (an eigenvalue) such that

$$-u'' + u = \lambda u$$

at all $x \neq 1/2$ with u not identically zero that satisfies

$$\begin{aligned} u'(0) &= 0 && (1)\\ u'(1) &= 0 && (2)\\ u'(1/2_-) &= u(1/2+) - u(1/2_-) && (3)\\ u'(1/2_+) &= u(1/2+) - u(1/2_-) && (4) \end{aligned}$$

where the subscripts $\pm$ mean right and left limits. It is not necessary to show code for this question but describe the formulation you used for this problem and how you implemented it in the MATLAB routine. *Note:* If you are not using MATLAB but can find a general boundary value problem solver in a library or want to write your own that is fine.

B2. We proved that on the interval $x \in [0,1]$ when $f \in C_2$ that

$$f(x) - I(x) = \frac{1}{2} f''(\xi) x(1-x)$$

where I is linear interpolation for some $\xi \in [0,1]$. In the argument ξ depends on x so we can consider $\xi(x)$ for a particular choice of ξ when there is more than one for some x values. Show that $\xi(x)$ can be chosen to be a measurable function. In particular, it can be chosen to be piecewise continuous (that is, continuous except at a countable number of points where right and left continuity holds).

B3. Consider a finite difference scheme for the periodic boundary value problem

$$-u'' + u = f(x)$$

based on values of u on a grid with spacing h. The values of f on the grid are known (the vector $\mathbf{F}$) and the values $\mathbf{U}$ on the grid are the unknowns. The values of $\mathbf{U}$ and $\mathbf{F}$ correspond to values of u'' on the

grid using the differential equation. Thus, we can construct a cubic approximation of u on each subinterval using Assignment #1, question A8. The equations for $\mathbf{U}$ are that the first derivatives at subinterval end-points must match between adjacent subintervals. Derive an explicit formula for these equations in terms of $\mathbf{U}$ and $\mathbf{F}$. Analyze the stability and accuracy of the resulting scheme.

Math 405/607, Fall 2014
Assignment #2 part A solutions

A1. Expand in Taylor series

$$F_{i+1} := f((i+1)h) = f(ih) + f'(ih)h + \frac{h^2}{2} f''(\xi_1)$$

for some $\xi_1 \in (ih, (i+1)h)$

So $$D_+ F_i := \frac{F_{i+1} - F_i}{h} = f'(ih) + \frac{h}{2} f''(\xi_1)$$

Similarly,

$$D_- F_i := \frac{F_i - F_{i-1}}{h} = f'(ih) + \frac{h}{2} f''(\xi_2)$$

for some $\xi_2 \in ((i-1)h, ih)$.

A2. $-u'' + u = x$ with $u(0)=0$ and $u'(1) = -u(1)$

homogeneous solutions

$$u_0(x) = Ae^x + Be^{-x}$$

particular solution (MUC)

$u_p(x) = a + bx$, a & b to be determined from the DE

plugging in, $a + bx = x$, so $u_p(x) = x$.

$u(x) = Ae^x + Be^{-x} + x$, A & B to be determined by the boundary conditions

$u'(x) = Ae^x - Be^{-x} + 1.$

$u(0) = 0 \Rightarrow A + B = 0 \; ; \; B = -A.$

$u'(1) = -u(1) \Rightarrow$

$$Ae + A/e + 1 = -(Ae - A/e + 1)$$

thus $A = -\frac{1}{e}$.

and $u(x) = -\frac{1}{e}e^x + \frac{1}{e}e^{-x} + x.$

(alternate form $u(x) = \frac{-2\sinh x}{e} + x$).

A3. I discretized the N points $x_j = jh$, with $h = 1/N$, using

$$\frac{2U_1}{h^2} - \frac{U_2}{h^2} + U_1 = f(h) \qquad j = 1.$$

standard $-D_2 U_j + U_j = f(jh) \quad j = 2, \ldots N-1$

At $Nh = 1$ I used the ghost point technique.

$$D_1 U_N = -U_N$$

$$\frac{U_{N+1} - U_{N-1}}{2h} = -U_N \Rightarrow U_{N+1} = U_{N-1} - 2hU_N.$$

thus $-D_2 U_N + U_N = f(1)$

$+\left(\frac{2}{h^2} + 1\right)U_N - \frac{1}{h^2}(U_{N+1} + U_{N-1}) = f(1)$ becomes

$$\left(\frac{2}{h^2} + 1 + 2/h\right) U_N - \frac{2}{h^2} U_{N-1} = F(1)$$

I tested the technique on problem A2, obtaining the following results:

N	Error $\|U - u\|_\infty$
8	1.13e-3
16	2.84e-4
32	7.01e-5
64	1.17e-6

↓ Increase factor 2

↓ decrease factor ≈ 4

Second order convergence is clearly seen.

A4. $-\hat{D}_2 \underline{U} + \underline{U} = \underline{F}$

$$\hat{D}_2 U_j = \frac{-U_{j-2} + 16 U_{j-1} - 30 U_j + 16 U_{j+1} - U_{j+1}}{12 h^2}$$

Taking $\underline{U} = \sum_{\alpha=0}^{N} \hat{U}_\alpha e^{i 2\pi \alpha j / N}$ and putting this form into the discrete equations, using the orthogonality of the DF Basis vectors gives

$$\lambda_\alpha \hat{U}_\alpha = \hat{F}_\alpha$$

where

$$\lambda_\alpha = +\frac{2}{12h^2}\cos(4\pi\alpha/N) - \frac{32}{12h^2}\cos(2\pi\alpha/N) + \frac{30}{12h^2} + 1.$$

Note that we want to show that

$$|\lambda_\alpha| \geq 1$$

for all α, h but a simple, absolute value argument can't be used here. Recall

$$\cos 2\theta = 2\cos^2\theta - 1.$$

If we take $\theta = 2\pi\alpha/N$, then

$$\lambda_\alpha = \frac{1}{12h^2}\left(4\cos^2\theta - 2 - 32\cos\theta + 30\right) + 1.$$

$$= \frac{1}{12h^2}\left(\underbrace{4(\cos\theta - 1)^2}_{\geq 0} - \underbrace{28\cos\theta + 28}_{\geq 0}\right) + 1$$

so $\lambda_\alpha \geq 1$, stability is shown.

AS. Note that $p(y) = y - 2y^3 + y^4$ is chosen so that $p(0) = p(1) = p''(0) = p''(1) = 0$ and $p'(0) = 1$.

So, $q(0) = q(1) = 0$ since $F(0) = C(0)$, $F(1) = C(1)$ and $p(0) = p(1) = 0$.

Thus, $q'(\xi_1) = 0$ for some $\xi_1 \in (0,1)$. (Rolle)

$q'(0) = 0$ since $p'(0) = 1 \Rightarrow$

$q''(\xi_2) = 0$ for some $\xi_2 \in (0, \xi_1) \subset (0,1)$

Now $q''(0) = q''(1) = 0$ since $F''(0) = C''(0)$, $F''(1) = C''(1)$ and $p''(0) = p''(1) = 0$ Using Rollé repeatedly, we see that

$$q'''(\xi_3) = q'''(\xi_4) = 0,$$

$\xi_3 \in (0, \xi_1)$, $\xi_4 \in (\xi_1, 1)$ and then

$$q''''(\xi) = 0.$$

$$q''''(y) = F''''(y) - 24\left[F'(0) - C'(0)\right].$$

so $q''''(\xi) = 0 \Rightarrow$

$$F'(0) - C'(0) = \frac{1}{24} F''''(\xi) \qquad (\star)$$

If $y \in [0,1]$ was obtained by scaling an interval of length h to y, then $\frac{dy}{dx} = 1/h$, $\frac{dx}{dy} = h$. $(\star)$ reads

$$\frac{df}{dy}(0) = \frac{dC}{dy}(0) = \frac{1}{24} \frac{d^4 f}{dy^4}(\xi)$$

which using the chain rule (affine maps)

$$\left(\frac{df}{dx}(0) - \frac{dC}{dx}(0)\right) h \overset{dx/dy}{\leftarrow} = \frac{h^4}{24} \frac{d^4 f}{dx^4}(\xi)$$

or $\left|\frac{df}{dx}(0) - \frac{dC}{dx}(0)\right| \leq \frac{h^3}{24} K_4$

(third order convergence).

MATH 405/607E, Fall 2014, Wetton
Assignment #3 - due Thursday, October 30

Instructions: Do all questions in part A. Do one question in part B. For students in Math 607, complete the project proposal in part C.

Part A Do all 5 questions. For part A questions, it is not necessary to be completely rigorous mathematically.

A1. Show that the DIRK-2 method in the class notes is second order accurate.

A2. Show that the DIRK-2 method in the class notes is A-stable.

A3. Consider the third order backward differentiation formula (BDF-3) method:

$$\frac{11}{6}U^n - 3U^{n-1} + \frac{3}{2}U^{n-2} - \frac{1}{3}U^{n-3} = kf(U^n, t_n)$$

Plot the stability region of the method. Determine whether the method is L-stable.

A4. Implement the Forward Euler scheme on the problem $u(t)$ with

$$\ddot{u} + u = 0$$

with initial conditions $u(0) = 1$, $\dot{u}(0) = 0$, written as a first order system. The exact solution is $u(t) = \cos t$. Observe the convergence of the scheme at $t = 8\pi$.

A5. In problem A4 above, the exact solution satisfies

$$\ddot{u}^2 + u^2 \equiv 1$$

for all time (this can be shown analytically). How does this quantity vary in time in your scheme from A4? Based on the stability region for FE and the eigenvalues of this problem written as a system, why would you expect this behaviour? Do any of the schemes considered in the lecture notes preserve this identity exactly? (either discuss why they don't or find one that does).

Part B. Do one question. Be as rigorous as you can for these questions.

B1. Consider the Forward Euler scheme applied to the general, scalar problem

$$\dot{u}(t) = f(u(t), t)$$

with $u(0) = u_0$ given initial data. Assume that $f \in C_2$ and that the solution is defined up to time $t = T$. Prove first order convergence of the scheme at time T.

B2. Show that the Radau IIA method in the class notes is indeed third order order, L-stable and A-stable.

B3. Implement an adaptive time stepping method based on Backward Euler (using Newton's method to solve the implicit problem) with a Forward Euler predictor as discussed in the class notes. The implementation should be as a function call, with inputs the initial conditions, a function that returns the values and derivatives of the right hand side, a vector of times at which the solution is required, and a local tolerance δ. Your output should be the solution vector at the desired times. Your implementation should be able to handle vector problems. Test your code out on several problems. Do not hand in your code, but show the errors for your test cases and how they behave as δ is decreased.

Part C: project proposal

Make a 2 page (strict maximum) proposal of your planned project, which can either be a longer proof or a computation of an applied problem. In your proposal, state clearly the theorem you want to prove or give some details of the application you are interested in. In either case, give a rough outline of your approach. Identify clearly three stages for the project: the first stage of a simple warm-up problem (plan zero), the second (plan A) which you are quite sure you can do, and then (plan B) the more difficult problems you will tackle if part A goes well. Your proposal is worth 5/20 of the marks for the project. Based on your proposal, I can verify that your proposed plan A results would give you a good mark if completed. As I mentioned at the beginning of the term, I would be happy if your project overlapped with your research work.

Math 405/607, Fall 2014, Assignment #3
Solutions, Part A.

(A1) Without loss of generality, we can consider $n=0 \to 1$.

$$U^* = U^0 + k\alpha f(U^*, \alpha k). \qquad (1).$$

This implicitly defines $U^*(k)$, which we will expand in a Taylor series. To get derivatives of $U^*(k)$ we will implicitly differentiate (1) wrt k.

$$\frac{\partial(1)}{\partial k} \Rightarrow \frac{dU^*}{dk} = \alpha f(U^*, \alpha k) + \alpha k\left(\frac{\partial f}{\partial u}\frac{dU^*}{dk} + \frac{\partial f}{\partial t}\alpha\right) \quad (2)$$

$$\frac{dU^*}{dk}(0) = \alpha f(U^0, 0).$$

$$\frac{\partial(2)}{\partial k} \Rightarrow \frac{d^2U^*}{dk^2} = 2\alpha\left(\frac{\partial f}{\partial u}\frac{dU^*}{dk} + \alpha\frac{\partial f}{\partial t}\right)\alpha +$$

$$\alpha k\left(f_{uu}\left(\frac{dU^*}{dk}\right)^2 + 2\alpha f_{ut}\frac{dU^*}{dk} + f_u\frac{d^2U^*}{dk^2} + \alpha^2 f_{tt}\right)$$

$$\frac{d^2U^*}{dk^2}(0) = 2\alpha^2\left(f_u(U^0,0)f(U^0,0) + f_t(U^0,0)\right).$$

So $U^* = U^0 + k\alpha f + k^2\alpha^2(f_u f + f_t) + O(k^3)$.

(all evaluated at $(U^0, 0)$)

From (1), we have

$$kf(U^*, \alpha k) = \frac{U^* - U^0}{\alpha} = kf + k^2\alpha(f_u f + f_t) + O(k^3) \qquad (3)$$

$$U^1 = U^0 + k\{\alpha f(U^{n+1}, k) + (1-\alpha) f(U^*, \alpha k)\}$$

consider the residual (local error) when we put the exact equation into this expression

$$u(k) - \{u(0) + k\{\alpha f(u(k), k) + (1-\alpha) f(u^*, \alpha k)\} \quad (4)$$

↑ when we use the exact value $U^0 = u(0)$

use Taylor expansions for $u(k)$ and $f(u(k),k) = \dot{u}(k)$ around $k=0$:

$$u(k) \approx u(0) + kf + \frac{k^2}{2}(f_u f + f_t) + O(k^3)$$

$$\dot{u}(k) \approx f + k(f_u f + f_t) + O(k^2)$$

and (3) in (4), collecting terms

$O(1)$: $u - u = 0$ ✓

$O(k)$: $f - \{\alpha f + (1-\alpha) f\} = 0$ ✓

$O(k^2)$: $\frac{1}{2}(f_u f + f_t) - \{\alpha (f_u f + f_t) + (1-\alpha)\alpha (f_u f + f_t)\}$

$$= \left(\frac{1}{2} - \alpha + (1-\alpha)\alpha\right)(f_u f + f_t) = 0 \quad ✓$$

↑

since $\alpha = 1 - \sqrt{2}/2$ is a root of this quadratic. Thus the local error is $O(k^3)$, the scheme has truncation error $O(k^2)$, it is second order accurate

Note: For completeness, I should have shown that the $O(k^3)$ term above is in general not zero. However, this was already more work than I intended for a part A question.

(A2) Apply DIRK-2 to $\dot{u} = \lambda u$:

$$U^* = U^n + \alpha k \lambda U^* \qquad (1)$$

$$U^{n+1} = U^n + k\{\alpha \lambda U^{n+1} + (1-\alpha)\lambda U^*\} \qquad (2)$$

(1) becomes $U^* = \dfrac{U^n}{1-\alpha z}$

(2) reads $U^{n+1} = U^n + \alpha z U^{n+1} + \dfrac{(1-\alpha) z U^n}{(1-\alpha z)}$.

so $U^{n+1} = \dfrac{1 + (1-\alpha)z/(1-\alpha z)}{(1-\alpha z)} U^n$

$$\downarrow$$

$$G(z) = \frac{1 - \alpha z + (1-\alpha) z}{(1-\alpha z)^2}$$

Since the expression for $G(z) = \dfrac{O(|z|)}{O(|z|^2)}$ as $|z| \to \infty$, the scheme is clearly A-stable.

(A3) Applying the scheme to $\dot{u} = \lambda u$ and considering $U^n = G^n$ we obtain

$$\frac{11}{6} G^3 - 3G^2 + \frac{3}{2} G - \frac{1}{3} = zG^3.$$

If z is on the boundary of the stability region then it corresponds to a value of G with $|G| = 1$, so $G = e^{i\theta}$ for some θ.

Taking a grid of θ points in $(0, 2\pi]$ and computing the corresponding z values

$$z = \frac{11/6\, e^{i3\theta} - 3e^{i2\theta} + \frac{3}{2} e^{i\theta} - 1/3}{e^{i3\theta}}$$

gives the figure on the following page. The stability region is the region outside this curve.

Since the stability region does not include the unstable left half plane, it is not an L-stable scheme.

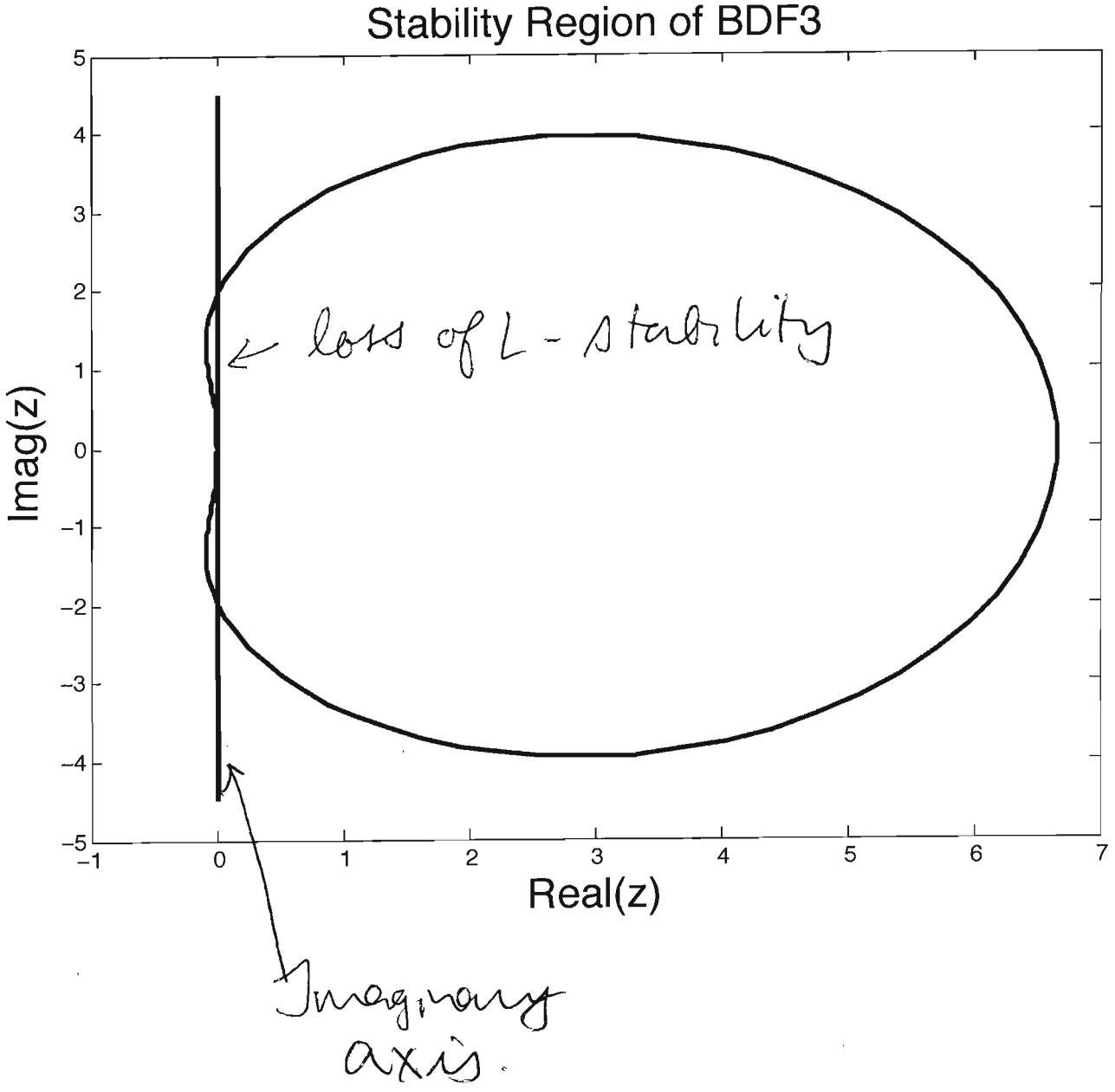
(A4) The problem is written as a first order system

$$\dot{u} = v, \quad u(0) = 1$$
$$\dot{v} = -u. \quad v(0) = 0$$

and computed out to time 8π with M time steps $k = 8\pi/M$

M	$U^M - 1$ ← $\cos(8\pi)$, exact value	$(U^M)^2 + (V^M)^2 - 1$ ← for (A5)
800	0.4837	1.2016
1600	0.2182	0.4840
3200	0.1037	0.2182
6400	0.0506	0.1037

M doubled, k halved → error approximately halved. This is first order convergence.

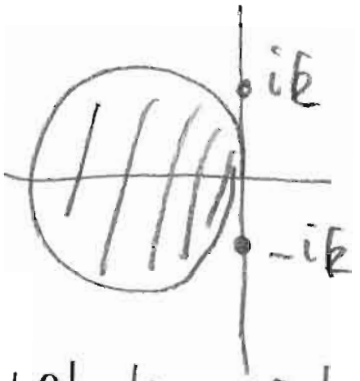

Stability Region of BDF3
Imag(z)
Real(z)
5
4
3
2
1
0
−1
−2
−3
−4
−5
−1
0
1
2
3
4
5
6
7
← loss of L-stability
Imaginary axis.

(A5) As seen in (A4), FE computed values of U^2+V^2 are always larger than exact values. This is expected, since

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

↑ eigenvalues $\pm i$

These correspond to k values $\pm ik$, which are always outside the stability region of FE:



and so $|G| = |1 \pm ik| > 1$ and discrete solutions grow in time.

If $U^2+V^2 \equiv 1$ is to be preserved exactly, then the time stepping scheme must have $|G| = 1$ when $z = \pm ik$. That is, the stability region boundary must be the imaginary axis. Only TR has this property.

You can check computationally that TR applied to

$$\dot{u} = v, \quad \dot{v} = -u \qquad u(0)=1,\ v(0)=0$$

keeps $u^2+v^2 \equiv 1$, or proceed analytically

$$u^{n+1} = u^n + \frac{k}{2}(v^n + v^{n+1})$$

$$v^{n+1} = v^n - \frac{k}{2}(u^n + u^{n+1})$$

$$\Rightarrow \begin{pmatrix} u^{n+1} \\ v^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -k/2 \\ k/2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & k/2 \\ k/2 & 1 \end{pmatrix} \begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

$$IA = \frac{1}{1+k^2/4} \begin{pmatrix} 1 & k/2 \\ -k/2 & 1 \end{pmatrix} \begin{pmatrix} 1 & k/2 \\ k/2 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1-k^2/4}{1+k^2/4} & \frac{k}{1+k^2/4} \\ \frac{-k}{1+k^2/4} & \frac{1-k^2/4}{1+k^2/4} \end{pmatrix}$$

Note that IA has the form $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$

with $a^2+b^2 = \dfrac{1-k^2/2+k^4/16+k^2}{(1+k^2/4)^2} = \dfrac{(1+k^2/4)^2}{(1+k^2/4)^2} = 1.$

Therefore, A is a 2D rotation matrix, so multiplying by A does not change the length of a vector. Since

$$\begin{pmatrix} U^n \\ V^n \end{pmatrix} = A^n \begin{pmatrix} U^0 \\ V^0 \end{pmatrix},$$

we have $(U^n)^2 + (V^n)^2 = (U^0)^2 + (V^0)^2 = 1$ for all n.

MATH 405/607E, Fall 2014, Wetton
Assignment #4 - due Thursday, November 13

Instructions: Do all questions in part A. Do one question in part B.

Part A Do all 5 questions. For part A questions, it is not necessary to be completely rigorous mathematically.

A1. Consider the heat equation for $u(x,t)$ with u 1-periodic in x:

$$u_t = u_{xx} \quad \text{with initial conditions } u(x,0) = u_0(x) \text{ given.}$$

The exact solution obeys a maximum principle, that is $u(x,t) \leq \max_x u_0(x)$ for all x and t. Show that this property also holds for finite difference discretizations of this problem when FE time stepping (with $k < h^2/2$) or BE time stepping is used.

A2. Consider the Lax Wendroff scheme applied to the one-way wave equation $u_t = u_x$:

$$U^{n+1} = U^n + kD_1U^n + \frac{k^2}{2}D_2U^n$$

where solutions are 1-periodic in x. Consider the scheme with $k = Ch$ with $0 < C < 1$. Show that it is second order accurate. Note that this is not a MOL scheme. *Hint:* You will have to differentiate the partial differential equation with time to match some of the terms.

A3. Show that the method above is stable as long as $k < h$. *Hint:* use von-Neumann analysis.

A4. Consider the following scheme for $\ddot{u} = f(u)$:

$$\begin{aligned} U^{n+1} &= U^n + k\left(V^n + \frac{k}{2}f(U^n)\right) \\ V^{n+1} &= V^n + \frac{k}{2}\left(f(U^n) + f(U^{n+1})\right). \end{aligned}$$

Show that the method is second order accurate. Note that it is a non-standard explicit scheme, specialized to this particular problem structure.

A5. Consider the following scheme for the 1D wave equation $u_{tt} = u_{xx}$:

$$\frac{1}{k^2}(U_j^{n+1} - 2U_j^n + U_j^{n-1}) = \frac{1}{h^2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n).$$

What time step restrictions are needed for the scheme to be stable?

A6. Describe a test problem with an exact solution for the 1D wave equation with nontrivial initial conditions for $u(x,0)$ and $u_t(x,0)$. Test out the scheme in A5 on your test problem and observe the convergence rate. You will need to find a strategy to initialize the value $\mathbf{U^1}$. Discuss your approach and results but do not submit your code.

Part B. Do one question. Be as rigorous as you can for these questions.

B1. Consider a vector of values $\mathbf{U}$ of a smooth function $u(x)$ on a grid with spacing h:

$$U_j = u(jh).$$

Let S be the operator that shifts values of $\mathbf{U}$ to the left

$$(S\mathbf{U})_j = U_{j-1}.$$

Show that for any p

$$\left(\frac{1}{h}\sum_{m=1}^{p}\frac{1}{m}(I-S)^m\right)\mathbf{U}$$

gives values of u' on the grid, with errors $O(h^p)$.

B2. Write a MATLAB function `adint` that does local adaptive quadrature. The function should have four arguments: the function name of the user defined function to be integrated, the end point values a and b of the interval of integration, and the desired accuracy δ of the result. The routine should have two outputs: the estimated value of the integral and the number of total function evaluations used. For this problem, e-mail the MATLAB code to me. You will be marked on how well the method works on test problems I pick. That is, whether it does achieve the target accuracy and whether it is reasonably efficient.

B3. Consider the heat equation $u_t = u_{xx}$ for $u(x,t)$ with u 1-periodic in x and smooth initial conditions $u_0(x)$. Show the convergence in maximum norm of a numerical scheme using standard, second order finite differences in space and FE time stepping with $k = \theta h^2$ with $\theta < 1/2$.

Math 405/607E, Assignment #4, part A solutions

(A1) Consider $M_n = \max_j U_j^n$.

(a) FE scheme

$$U_j^{n+1} = U_j^n + \frac{k}{h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right)$$

$$= \left(\frac{k}{h^2}\right)U_{j-1}^n + \left(1 - \frac{2k}{h^2}\right)U_j^n + \frac{k}{h^2}U_{j+1}^n$$

(positive if $k < h^2/2$)

$$\leq \left(1 - \frac{2k}{h^2} + \frac{k}{h^2} + \frac{k}{h^2}\right)M_n = M_n.$$

(convex combination)

Taking the maximum over j, we have

$$M_{n+1} \leq M_n.$$

By induction $M_n \leq M_0$ for all $n \geq 0$, so

$$U_j^n \leq \max_j U_j^0 \leq \max_x u_0(x).$$

(b) BE scheme.

$$-\frac{k}{h^2}U_{j-1}^{n+1} - \frac{k}{h^2}U_{j+1}^{n+1} + \left(1 + \frac{2k}{h^2}\right)U_j^{n+1} = U_j^n \qquad (1)$$

Consider (1) for an index j such that $U_j^{n+1} = M_{n+1}$

Then $-U_{j-1}^{n+1} - U_{j+1}^{n+1} + 2U_j^{n+1} \geq 0$, so (1) implies

$$M_{n+1} := U_j^{n+1} \le U_j^n \le M_n.$$

The result follows from $M_{n+1} \le M_n$ as in (a) above.

(A2) Consider the truncation error, with u the exact solution

$$\tau = \frac{1}{k}\left(u^{n+1} - u^n - k D_1 u^n - \frac{k^2}{2} D_2 u^n\right)$$

↓ Taylor series.

$$\tau = \frac{1}{k}\left\{\left(u^n + k u_t^n + \frac{k^2}{2} u_{tt}^n + O(k^3)\right) - u^n - k\left(u_x^n + \frac{h^2}{6} u_{xxx}^n + O(h^4)\right) - \frac{k^2}{2}\left(u_{xx}^n + \frac{h^2}{12} u_{xxxx}^n + O(h^4)\right)\right\}$$

Under the scaling $k = Ch$, $O(h) = O(k)$ so we can simplify the expression above to

$$\tau = \frac{1}{k}\left((u^n - u^n) + k(u_t^n - u_x^n) + \frac{k^2}{2}(u_{tt} - u_{xx}) + O(k^3)\right)$$

$u_t^n - u_x^n$: zero by equation

$u_{tt} - u_{xx}$: starting with $u_t = u_x$ we can differentiate

$$u_{tt} = u_{xt} = u_{tx} = \ldots = u_{xx}$$

↑ $u_t = u_x$

So $\tau = O(k^2)$, second order accurate ✓.

(A3) Consider $U_j^n = \lambda_\alpha^n e^{2\pi i j \alpha h}$ (2)

Note: this is a shortcut to doing the eigenanalysis of the matrix $I + k D_1 + \frac{k^2}{2} D_2$ since we know that the eigenvectors will be the discrete Fourier vectors.

Put the form (2) into the discrete equations and simplify to

$$\lambda_\alpha = 1 + i\frac{k}{h}\sin(\underbrace{2\pi\alpha h}_{\theta}) - \frac{k^2}{h^2 2}\left(2 - 2\cos(\underbrace{2\pi\alpha h}_{\theta})\right).$$

For stability, we want $|\lambda_\alpha| \le 1$, use $\beta = \frac{k}{h}$:

$$|\lambda_\alpha|^2 = \left(1 - \beta^2 + \beta^2\cos\theta\right)^2 + (\beta\sin\theta)^2$$

$$= 1 - 2\beta^2 + \beta^4 + 2\beta^2\cos\theta - 2\beta^4\cos\theta + \beta^4\cos^2\theta + \beta^2\sin^2\theta.$$

Let $c = \cos\theta$ and use $\sin^2\theta = 1 - \cos^2\theta$,

$$g(c,\beta) = |\lambda_\alpha|^2 = 1 - \beta^2 + \beta^4 + \beta^2(\beta^2 - 1)c^2 + 2(1-\beta^2)\beta^2 c.$$

Consider $g(c,\beta)$ for fixed β, $-1 \le c \le 1$.

$$\frac{dg}{dc} = 2\beta^2(\beta^2 - 1)(c - 1)$$

So the only critical point is at $c = 1$ (also a boundary value). Thus, the maximum value of g for fixed β must occur at $c = \pm 1$.

$$g(1,\beta) = 1$$

$$g(-1,\beta) = 1 - 4\beta^2(1-\beta^2).$$

Note that $g(-1,\beta) \le 1$ for $\beta \le 1$, but $g(-1,\beta) > 1$ for $\beta > 1$.

Thus, the method is stable ($|\lambda_\alpha| \le 1$ for all α) if $\beta \le 1$, that is, $k < h$.

(A4) Exact solution $v = \dot{u}$, $\ddot{u} = f(u)$. Put the exact solution into the discrete equations to determine the truncation error.

$$\tau_1 = \frac{1}{k}\left(u^{n+1} - u^n - kv^n - \frac{k^2}{2} f(u^n)\right)$$

$$= \frac{1}{k}\Big(\big(u^n + k\dot{u}^n + \frac{k^2}{2}\ddot{u}^n + O(k^3)\big) - u^n - k\dot{u}^n - \frac{k^2}{2}\ddot{u}^n\Big) = O(k^2) \quad \checkmark$$

$$\tau_2 = \frac{1}{k}\left(v^{n+1} - v^n - \frac{k}{2}\left(f(u^n) + f(u^{n+1})\right)\right)$$

$$= \frac{1}{k}\Big(\big(v^n + k\dot{v}^n + \frac{k^2}{2}\ddot{v}^n\big) + O(k^3) - \frac{k}{2}f^n - \frac{k}{2}\big(f^n + k\dot{f}^n + O(k^2)\big)\Big)$$

$$= \frac{1}{k}\Big(\big(\dot{u}^n + k\ddot{u}^n + \frac{k^2}{2}\dddot{u}^n\big) + O(k^3) - \frac{k}{2}\ddot{u}^n - \frac{k}{2}\ddot{u}^n - \frac{k^2}{2}\dddot{u}^n + O(k^3)\Big)$$

$$= O(k^2) \quad \checkmark$$

Thus, the scheme is second order accurate.

(A5) Consider the behaviour of a DF vector in time, that is consider U_j^n of the form

$$U_j^n = \hat{U}_\alpha^n \, e^{2\pi i \alpha j h}.$$

Putting this form into the discrete equations gives, using $\beta = k/h$.

$$\hat{U}_\alpha^{n+1} - 2\left(1 + \beta^2(\cos\theta - 1)\right)\hat{U}_\alpha^n + \hat{U}_\alpha^{n-1} = 0.$$

As in the study of multistep methods, this constant coefficient, homogeneous, linear, 2nd order difference equation has solution

$$U_\alpha^n = A G_1^n + B G_2^n$$

where $G_{1,2}$ are roots of

$$G^2 - 2\left(1 + \beta^2(\cos\theta - 1)\right)G + 1 = 0.$$

Note that $G_1 G_2 = 1$ → so the roots come in reciprocal pairs. If they are real and different, then either $|G_1| > 1$ or $|G_2| > 1$. If the roots are complex then $G_{1,2}$ are a conjugate pair with $|G_1| = |G_2| = 1$. Thus, $|G_1| = |G_2| = 1$ (stability) when

$$4\left(1 + \beta^2(\cos\theta - 1)\right)^2 - 4 < 0 \qquad \text{(discriminant of quadratic)}.$$

That is,

$$\left[1+\underbrace{\beta^2(\cos\theta-1)}_{\leq 0,\ \geq -2\beta^2}\right]^2 \leq 1 \quad \text{for all } \theta\in[-\pi,\pi].$$

Thus, this condition is satisfied if $\beta<1$, thus we have a stability restriction $k<h$.

(A6) I take a test solution

$$u(x,t) = e^{\cos(2\pi(x+t))}$$

on a domain $x\in[0,1]$ with periodic boundary conditions, so

$$u(x,0) = e^{\cos(2\pi x)}$$

and $u_t(x,0) = 2\pi \sin(2\pi x)\, e^{\cos(2\pi x)}$.

I initialize with

$$U_j^0 = e^{\cos(2\pi j h)} \qquad \text{(exact)}$$

and approximate

$$U_j^1 = e^{\cos(2\pi j h)} + k\,\underbrace{2\pi \sin(2\pi j h)\, e^{\cos(2\pi j h)}}_{u_t}$$

(linear Taylor approx)

I compute with $k \approx 0.9h$ to time 1 and compare to the exact solution

N	Error (max norm)
16	4.6e-2
32	9.0e-3
64	2.2e-3
128	5.1e-4

↓ Second order convergence, error $\sim O(h^2) = O(k^2)$.

MATH 405/607E, Fall 2014, Wetton
Assignment #5 - due Thursday, November 27

Instructions: Do all questions in part A. Do one question in part B.

Part A Do all 4 questions. For part A questions, it is not necessary to be completely rigorous mathematically.

A1. Consider the following problem for $u(x)$, 1-periodic in x:

$$u'''' + u = f(x)$$

and its discretization on a uniform grid with spacing h:

$$D_4\mathbf{U} = \mathbf{F}$$

where $D_4 = D_2D_2$, explicitly:

$$D_4U_j = \frac{1}{h^4}\left(U_{j-2} - 4U_{j-1} + 6U_j - 4U_{j+1} + U_{j+2}\right).$$

Show (using von Neumann analysis) that the discretization is stable in scaled Euclidean (l_2) norm.

A2. Show that if Jacobi iterations are applied to the discretization in #A1 above, that undamped ($\omega = 1$) iterations do *not* converge, but that damped iterations with $\omega = 1/2$ do converge and have a smoothing property.

A3. Find the eigenvalues analytically of the two grid operator M_2 from the Day 19 notes, page 6. Note that the Discrete Fourier vectors are not (quite) the eigenvectors of that matrix.

A4. Implement the discretization of the 2D problem described in pages 1-3 of the Day 18 notes. Submit a contour plot of the solution you compute with $N = 64$ and

$$f = e^{\cos(2\pi x)+y}\left(-y^2 - 3y + 4\pi^2(\sin^2(2\pi x) - \cos(2\pi x))(y - y^2)\right)$$

Part B. Do one question. Be as rigorous as you can for these questions.

B1. Find a third order quadrature method on the unit equilateral triangle Ω using three points. That is, find 3 points (x_i, y_i) and weights w_i such that

$$\int_\Omega p(x,y)dA = \sum_{i=1}^{3} w_i p(x_i, y_i)$$

for all polynomials $p(x, y)$ of degree 2. *Hint:* look for some symmetry and assume that the weights are all the same (one third of the triangle area). *Bonus points:* find a fourth order quadrature method for this domain (that is, exact for polynomials up to degree 3) with positive weights. You will need more than 3 points to do this).

B2. Implement (code yourself) a two-grid solver for #A4 and show that it converges independent of h. *Bonus points:* implement a multi-grid solver for this problem and give evidence that it converges independent of the finest grid spacing h.

B3. Use the MATLAB `pdetool` to compute the solution of

$$\Delta u = -1$$

with $u = 0$ on the boundary of the 2D domain that is the union of the unit square $[0, 1]^2$, the circle of radius 1/2 centred at (1,1/2) and the triangle with vertices (0,0), (0,1) and (-1,1/2). Your answer to this question is the contour plot of your approximate solution.

Math 405/607, Fall 2014, assignment #5
part A solutions.

(A1) the symbol of $\mathbb{D}_4 + \mathbb{I}$ is

$$\frac{1}{h^4}(2\cos 2\theta - 8\cos\theta + 6) + 1 := \lambda_\alpha. \qquad (1)$$

where $\theta = 2\pi\alpha h$ as usual. Consider.

$$2\cos 2\theta - 8\cos\theta + 6$$
$$= 2(2\cos^2\theta - 1) - 8\cos\theta + 6 \qquad \downarrow\ x = \cos\theta$$
$$\Rightarrow \quad f(x) = 4 + 4x^2 - 8x \qquad x \in [-1, 1]$$
$$f(1) = 0, \quad f(-1) = 16.$$
$$f'(x) = 8x - 8, \text{ so } f'(x) = 0 \text{ at } x = 1,$$
already considered.

Thus $f(x) \geq 0$ on the interval, so

$$\lambda_\alpha = \frac{1}{h^4} f(\cos\theta) + 1 \geq 1$$

so $\frac{1}{\lambda_\alpha} \leq 1$ independent of α and h, so $\|A^{-1}\|_2 \leq 1$, the discretization is stable.

(A2) Jacobi iterations, undamped ($\omega = 1$)

$$\frac{1}{h^4}\left(U_{j-2}^{(m)} - 4U_{j-1}^{(m)} + 6U_j^{(m+1)} - 4U_{j+1}^{(m)} + U_{j+2}^{(m)}\right) + U_j^{(m+1)} = F_j.$$

$$\left(\frac{6}{h^4}+1\right)U_j^{(m+1)} = -\frac{1}{h^4}\left(U_{j-2}^{(m)} - 4U_{j-1}^{(m)} - 4U_{j+1}^{(m)} + U_{j+2}^{(m)}\right) + F_j.$$

$\frac{6}{h^4}+1 = \frac{6+h^4}{h^4}$

$$U_j^{(m+1)} = -\frac{1}{6+h^4}\left(U_{j-2}^{(m)} - 4U_{j-1}^{(m)} - 4U_{j+1}^{(m)} + U_{j+2}^{(m)}\right) + \frac{h^4}{6+h^4}F_j.$$

Subtracting the exact solution, we get

$$E_j^{(m+1)} = -\frac{1}{6+h^4}\left(E_{j-2}^{(m)} - 4E_{j-1}^{(m)} - 4E_{j+1}^{(m)} + E_{j+2}^{(m)}\right).$$

[could have got this from the general expression

$$\underline{E}^{(m+1)} = -\left(D^{-1}(L+U)\right)\underline{E}^{(m)}\ ]$$

symbol $S_\alpha = \frac{1}{6+h^4}\left(-2\cos(2\theta) + 8\cos(\theta)\right)$.

when $\theta = \pm\pi$ (attained for $\alpha = \pm N/2$),

$$S_\alpha = \frac{-10}{6+h^4}, \quad |S_\alpha| > 1$$

↑

Iterations do not converge.

Damped Jacobi iterations with $\omega = 1/2$ have

Symbol

$$\frac{1}{2}S\alpha + \frac{1}{2} = \frac{-\cos(2\theta) + 4\cos\theta + 3 + h^4/2}{6 + h^4} \qquad (1)$$

Considering $|\alpha| \geq N/4$ (oscillatory modes) is equivalent to considering

$$\cos\theta \;, |\theta| > \pi/2, \quad \text{so } X = \cos\theta \in [-1, 0].$$

$$\frac{1}{2}S\alpha + \frac{1}{2} = \frac{f(\cos\theta) + h^2/2}{6 + h^4}, \quad f(x) = 4 - 2x^2 + 4x$$

where, using the same argument as in A1 it can be shown that

$$-2 \leq f(\cos\theta) \leq 4 \qquad \text{for } \cos\theta \in [-1, 0].$$

Considering (1) with this result, the smoothing factor lies in the interval

$$\frac{-2 + h^4/2}{6 + h^4} \leq \frac{1}{2}S\alpha + \frac{1}{2} \leq \frac{4 + h^4/2}{6 + h^4} \leq \frac{4 + 2h^4/3}{6 + h^4} \leq 2/3.$$

$$\uparrow$$

$$X = \cos\theta \in [-1, 0]$$

Since $\left| \frac{-2 + h^4/2}{6 + h^4} \right| \leq 2/3$ for $h < 1$, we have a smoothing property guaranteed with rate $2/3$.

(A3) This problem was more technically difficult than I intended

Preliminaries: Consider half of the DF vectors on the fine grid of N (even) points with spacing $h = 1/N$.

$$[\underline{F}_\alpha]_j = e^{2\pi i j \alpha h}, \quad \alpha = 0, \ldots N/2 - 1.$$

Consider the other half indexed in the form $\alpha' = \alpha - N/2$,

$$[\underline{F}_{\alpha'}]_j = e^{2\pi i j (\alpha - N/2) h} = (-1)^j [\underline{F}_\alpha]_j.$$

Above, $j = 0, \ldots N-1$ (N fine grid points).

Consider also the DF α vector on the coarse grid, using grid point index ℓ:

$$[G_\alpha]_\ell = e^{2\pi i \ell \alpha (2h)}, \quad \alpha = 0, \ldots N/2 - 1, \quad \ell = 0, \ldots N/2 - 1.$$

($2h$: coarse grid)

It will be shown that eigenvectors of M_2 have the form

$$\underline{V} = a\, \underline{F}_\alpha + b\, \underline{F}_{\alpha'} \qquad (o)$$

where the eigenvectors are eigenvalues of $N/2$, 2×2 matrices B_α and $\begin{bmatrix} a \\ b \end{bmatrix}$ are the corresponding eigenvectors.

Each B_α gives 2 eigenvalues for $\alpha = 0, \ldots N/2-1$, so this gives all N eigenvalues of M_2.

Aside: We take $\theta = 2\pi\alpha h$ as usual, and so $\theta \in [0, \pi)$ with our choice of $\alpha = 0, \ldots N/2-1$. For $\theta \geq 0$ but small, these are modes for which $\underline{F_\alpha}$ is smooth on the fine grid. For $\theta \simeq \pi$, $\underline{F_{\alpha'}}$ is smooth on the fine grid. We'll follow this aside through the algebra below.

$$S \underline{F_\alpha} = S_\alpha \underline{F_\alpha}, \quad S_\alpha := \frac{2}{3}\left(\frac{\cos\theta}{1+h^2/2}\right) + 1/3.$$

from Day 19 notes.

$$S \underline{F_{\alpha'}} = S_{\alpha'} \underline{F_{\alpha'}}, \quad S_{\alpha'} := \frac{2}{3}\left(\frac{\cos(\theta-\pi)}{1+h^2/2}\right) + 1/3$$

$$= -\frac{2}{3}\left(\frac{\cos\theta}{1+h^2/2}\right) + 1/3.$$

Aside: $S_\alpha \approx 1$ for θ small

$S_{\alpha'} \approx 1$ for $\theta \approx \pi$

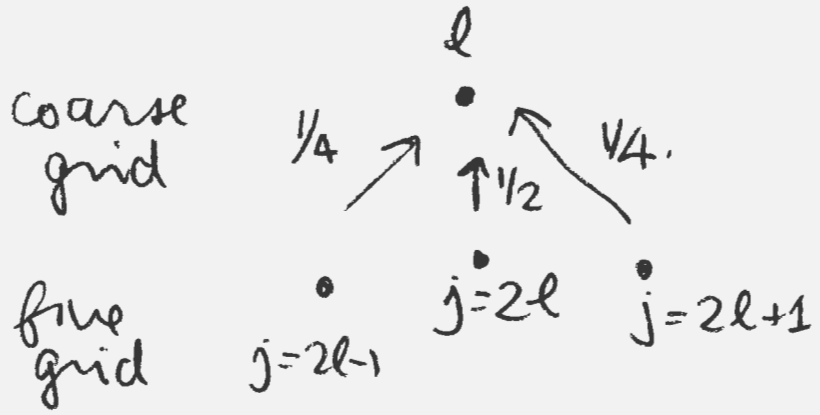
this notates the intuition that smoothing does not reduce the error in smooth modes.

$$A_h \underline{F_\alpha} = \lambda_\alpha \underline{F_\alpha}, \quad \lambda_\alpha = 1 + \frac{2(1-\cos\theta)}{h^2}$$

(earliest von N analysis)

$$A_h \underline{F_{\alpha'}} = \lambda_{\alpha'} \underline{F_\alpha}, \quad \lambda_{\alpha'} = \frac{1 + 2(1+\cos\theta)}{h^2}$$

Consider $(I_R \underline{F_\alpha})_\ell$, following the picture below



$$(I_R \underline{F_\alpha})_\ell = \frac{1}{4}\left([\underline{F_\alpha}]_{2\ell-1} + [\underline{F_\alpha}]_{2\ell+1}\right) + \frac{1}{2}[F_\alpha]_{2\ell}$$

$$= \underbrace{\frac{1}{2}(1+\cos\theta)}_{r_\alpha} e^{2\pi i (2\ell)\alpha h}$$

move 2 over

$$= r_\alpha [\underline{G_\alpha}]_\ell$$

similarly

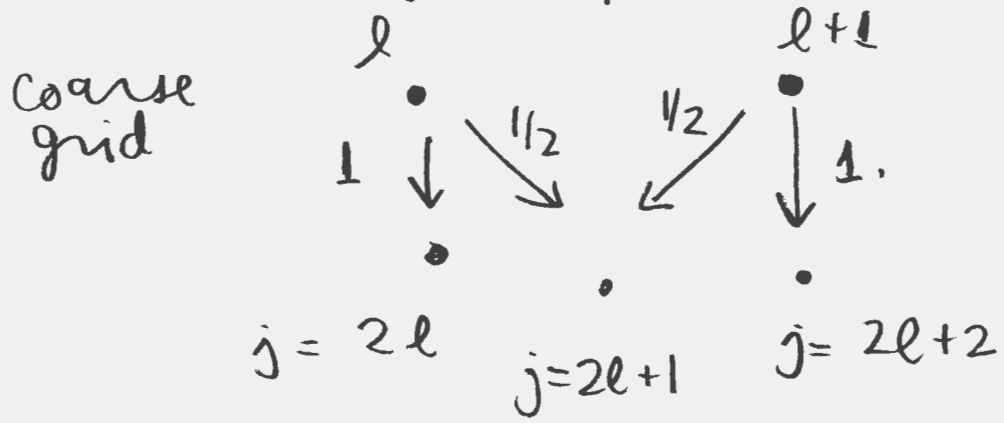
$$I_R \underline{F_{\alpha'}} = r_{\alpha'} \underline{G_\alpha}, \quad r_{\alpha'} = \frac{1}{2}(1-\cos\theta)$$

aside: Note that I_R preserves the size of smooth vectors, which weren't damped by S.

$$[A_{2h}]^{-1}\underline{G_\alpha} = \frac{1}{\mu_\alpha}\underline{G_\alpha}$$

$$\mu_\alpha = 1 + \frac{2(1-\cos(2\theta))}{4h^2}$$ ← note this is just λ_α with $h \to 2h$.

$(2R^T \underline{G_\alpha})$ is a vector on the fine grid, computed differently depending on whether j is an even or odd grid point:



$$(2R^T \underline{G_\alpha})_j = \begin{cases} [\underline{G_\alpha}]_\ell = [\underline{F_\alpha}]_j & \text{if } j = 2\ell \text{ (even)} \\ \frac{1}{2}\left([\underline{G_\alpha}]_\ell + [\underline{G_\alpha}]_{\ell+1}\right) & \text{if } j = 2\ell+1 \text{ (odd)} \\ \quad = \frac{1}{2}\left([\underline{F_\alpha}]_{j-1} + [\underline{F_\alpha}]_{j+1}\right) & \\ \quad = \cos\theta\,[\underline{F_\alpha}]_j & \end{cases}$$

We can write this expression without the "if" statement as follows:

$$(2R^T G_\alpha)_j = \tfrac{1}{2}(1+\cos\theta)(F_\alpha)_j + (-1)^j (1-\cos\theta)/2 [F_\alpha]_j$$

$$= \underbrace{\tfrac{1}{2}(1+\cos\theta)}_{r_\alpha}(F_\alpha)_j + \underbrace{\tfrac{1}{2}(1-\cos\theta)}_{r_{\alpha'}}(F_{\alpha'})_j$$

Aside: G_α prolongates to F_α (mostly) when $\theta \approx 0$ and to $F_{\alpha'}$ when $\theta \approx -\pi$.

Consider the coarse grid correction operator acting on F_α

$$(I - 2R^T A_{2h}^{-1} R A) F_\alpha$$

$$A_{2h}^{-1} R A F_\alpha = \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha G_\alpha$$

So $2R^T A_{2h}^{-1} R A F_\alpha = \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha^2 F_\alpha + \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha r_{\alpha'} F_{\alpha'}$

and

$$(I - 2R^T A_{2h}^{-1} R A) F_\alpha =$$

$$\underbrace{\left(1 - \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha^2\right) F_\alpha - \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha r_{\alpha'}}_{\uparrow} \qquad (1)$$

Aside: these are ≈ 0 for θ small, so the coarse grid correction fixes the error in smooth

components. ✓

Similarly

$$(\mathbb{I} - 2\mathbb{R}^T \mathbb{A}_{2n}^{-1} \mathbb{R}\mathbb{A})\ \underline{F_{\alpha'}} \qquad (2)$$

$$= \left(1 - \frac{\lambda_{\alpha'}}{\mu_\alpha} r_{\alpha'}^2\right) \underline{F_{\alpha'}} - \frac{\lambda_{\alpha'}}{\mu_\alpha} r_\alpha r_{\alpha'} \underline{F_\alpha}$$

Using (1) & (2) we see that $\mathbb{M}_2 \underline{V}$ where $\underline{V}$ is from (6) on page 4, has the form

$$\mathbb{M}_2 \underline{V} = c\,\underline{F_\alpha} + d\,\underline{F_{\alpha'}}$$

where

$$\begin{bmatrix} c \\ d \end{bmatrix} = \mathbb{B}_\alpha \begin{bmatrix} a \\ b \end{bmatrix} \qquad (3)$$

and $\mathbb{B}_\alpha = \begin{bmatrix} S_\alpha & 0 \\ 0 & S_{\alpha'} \end{bmatrix} \begin{bmatrix} 1 - \frac{\lambda_\alpha}{\mu_\alpha} r_\alpha^2 & -\frac{\lambda_\alpha}{\mu_\alpha} r_\alpha r_{\alpha'} \\ -\frac{\lambda_{\alpha'}}{\mu_\alpha} r_\alpha r_{\alpha'} & 1 - \frac{\lambda_{\alpha'}}{\mu_\alpha} r_{\alpha'}^2 \end{bmatrix} \times$

$$\begin{bmatrix} S_\alpha & 0 \\ 0 & S_{\alpha'} \end{bmatrix}.$$

Thus $\underline{V}$ will be an eigenvector of $\mathbb{M}_2$ with corresponding eigenvalue δ if $\begin{bmatrix} a \\ b \end{bmatrix}$ is an eigenvector of $\mathbb{B}_\alpha$ with eigenvalue δ.

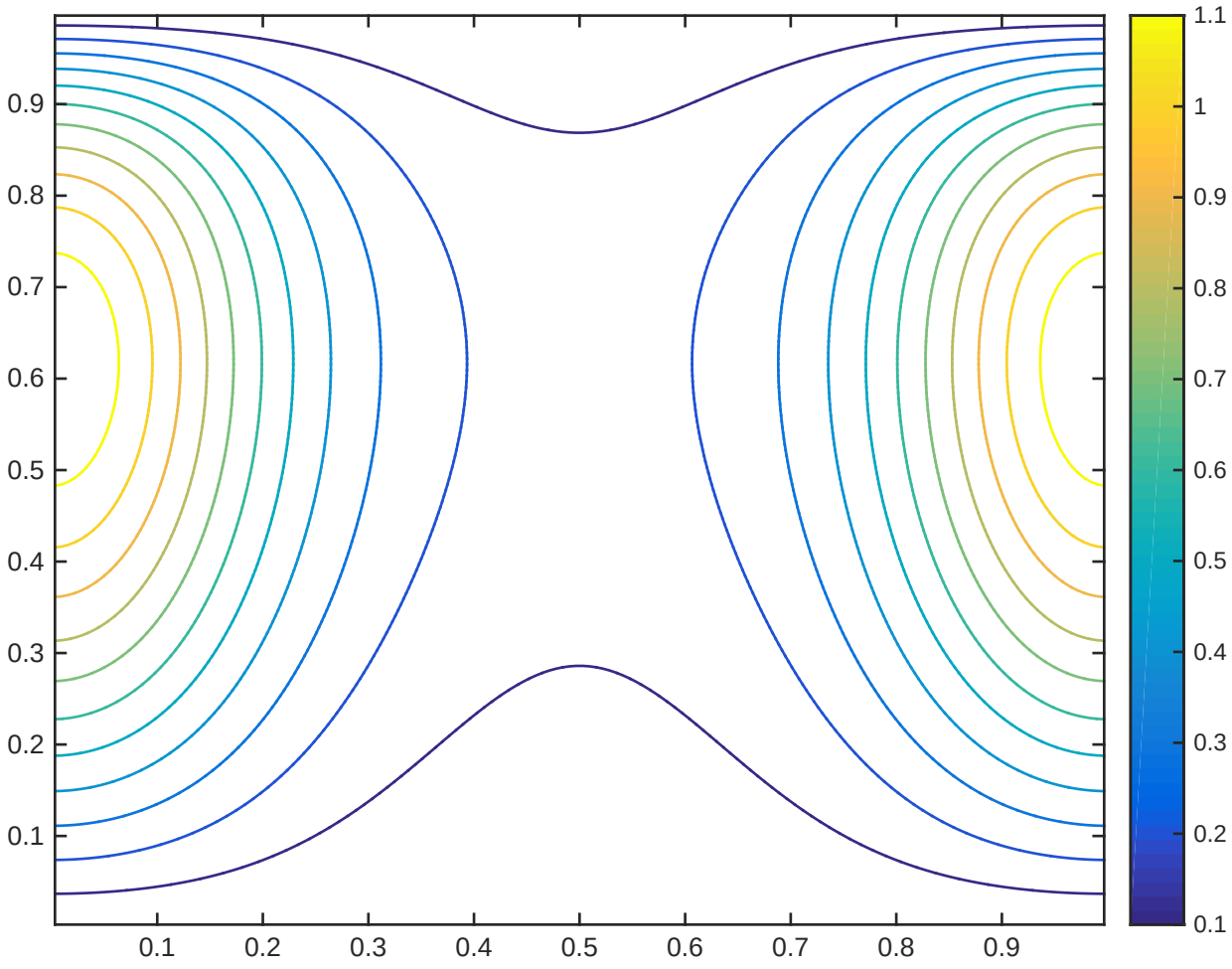
I stopped here with analytic progress, but ran a MATLAB script on (3) for various θ and got results matching the eigenvalues computed for the full M_2 matrix shown in class. (some eigenvalues ≈ 0, others ≈ 0.1).

Note: It appears that as $h \to 0$ and excluding the $\alpha = 0$ ($\theta = 0$) case, B_α has one zero eigenvalue for every α. With $h > 0$ this is only approximately true.

(A4) I posted my code for this problem, in script a5q4.m. Note that the problem has the exact solution

$$u(x,y) = e^{\cos(2\pi x) + y} \cdot (y - y^2)$$

Note that my code uses N^2 unknowns on a cell centered grid.


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Name: _____ Student #: _____

Math 405/607 Midterm, Fall 2014

Instructions:

- No notes, no calculators.
- Show all work. Justify your answers.
- Do all four questions in part A. Do one question in part B.
- All questions worth 5 marks, total 25 marks.

part A: do all questions

Questions A1 and A2 concern the following non-standard quadratic approximation on the reference interval$[0, 1]$:

$$f(y) \approx Q(y) := f(0)(1 - y^2) + f'(0)(y - y^2) + f(1)y^2$$

A1. For the approximation above

(a) [2 marks] If $f(0) = 1$, $f'(0) = 2$ and $f(1) = 4$ what is the interpolant approximation $Q(1/2)$?

(b) [3] What properties specify the polynomial $p(y) = y - y^2$ in the expression above?

$$f(y) \approx Q(y) := f(0)(1-y^2) + f'(0)(y-y^2) + f(1)y^2$$

A2. Show that for every $b \in (0,1)$,

$$f(b) - Q(b) = \frac{1}{6}(b^2 - b^3)f'''(\xi)$$

for some $\xi \in (0,1)$.

A3. Consider the nonlinear problem of $u(x)$, with u 1-periodic:

$$-u'' + u + u^3 = f(x)$$

and its discretization on a uniform grid of N points with spacing h:

$$-D_2 U_j + U_j + U_j^3 = f(jh).$$

What are the entries of the $N \times N$ Jacobian matrix that corresponds to this system?

A4. The second order approximation of the fourth derivative on a uniform-grid with spacing h is given by

$$D_4 U_j = \frac{U_{j-2} - 4U_{j-1} + 6U_j - 4U_{j+1} + U_{j+2}}{h^4}$$

Show that the eigenvalues of D_4 on periodic problems are non-negative. *Hints:* use von Neumann analysis and the identity $\cos(2\theta) = 2\cos^2\theta - 1$.

Part B: do either B1 or B2

B1. Consider the nonstandard quadrature rule

$$\int_{-1}^{1} f(x)dx \approx w_1 f(x_1) + w_1 f(-x_1) + w_2 f'(x_2) - w_2 f'(x_2).$$

It is nonstandard since in involves derivative values. Find relationships between x_1, x_2, w_1 and w_2 so that the quadrature is sixth order accurate.

B2. Consider standard quadratic interpolation on the reference interval $y \in (-1, 1)$:

$$f(y) \approx Q(y) := f(-1)(y^2 - y)/2 + f(0)(1 - y^2) + f(1)(y^2 + y)/2$$

(a) [2 marks] Show with an example that even if $f(-1)$, $f(0)$, and $f(1)$ are all positive, $Q(y)$ can be negative for some values of y. (The right picture would be enough here).

(b) [3] How could you modify the interpolation above so that if $f(-1)$, $f(0)$, and $f(1)$ were all positive, $Q(y)$ would be positive for all y, while maintaining the accuracy of quadratic interpolation? *Hint:* what function always has positive values?

Blank page, more space for part B answers

Name: Solutions. Student #: _____

Math 405/607 Midterm, Fall 2014

Instructions:

- No notes, no calculators.
- Show all work. Justify your answers.
- Do all four questions in part A. Do one question in part B.
- All questions worth 5 marks, total 25 marks.

part A: do all questions

Questions A1 and A2 concern the following non-standard quadratic approximation on the reference interval[0, 1]:

$$f(y) \approx Q(y) := f(0)(1-y^2) + f'(0)(y-y^2) + f(1)y^2$$

A1. For the approximation above

(a) [2 marks] If $f(0) = 1$, $f'(0) = 2$ and $f(1) = 4$ what is the interpolant approximation $Q(1/2)$?

(b) [3] What properties specify the polynomial $p(y) = y - y^2$ in the expression above?

(a) $f(\frac{1}{2}) \approx Q(\frac{1}{2}) = 1\left(1-(\frac{1}{2})^2\right) + 2\left(\frac{1}{2}-(\frac{1}{2})^2\right) + 4(\frac{1}{2})^2$

$= \frac{3}{4} + \frac{2}{4} + 1 = 9/4$

(b) p quadratic, $p(0)=0$, $p(1)=0$, $p'(0)=1$.

$$f(y) \approx Q(y) := f(0)(1-y^2) + f'(0)(y-y^2) + f(1)y^2$$

A2. Show that for every $b \in (0,1)$,

$$f(b) - Q(b) = \frac{1}{6}(b^2 - b^3) f'''(\xi)$$

for some $\xi \in (0,1)$.

Consider $g(y) = F(y) - Q(y) - \dfrac{y^2 - y^3}{b^2 - b^3}\left(F(b) - Q(b)\right)$

$g(0) = 0$, $g(1) = 0$, $g(b) = 0$, so (Rolle)

$g'(\xi_1) = 0$, $g'(\xi_2) = 0$, $\quad 0 < \xi_1 < b < \xi_2 < 1$.

$$g'(y) = F'(y) - Q'(y) - \frac{(2y - 3y^2)}{b^2 - b^3}\left(F(b) - Q(b)\right)$$

So $g'(0) = 0$ also.

$g' = 0$ at $0, \xi_1, \xi_2$ distinct points, multiple Rolle...

$g'''(\xi) = 0$ for some $\xi \in (0,1)$.

$$g'''(y) = F'''(y) - \frac{6}{b^2 - b^3}\left(F(b) - Q(b)\right)$$

The result follows.

A3. Consider the nonlinear problem of $u(x)$, with u 1-periodic:

$$-u'' + u + u^3 = f(x)$$

and its discretization on a uniform grid of N points with spacing h:

$$-D_2 U_j + U_j + U_j^3 = f(jh).$$

What are the entries of the $N \times N$ Jacobian matrix that corresponds to this system?

Nonlinear system $N_i = -\frac{U_{i+1}}{h^2} + \left(\frac{2}{h^2}+1\right) U_i - \frac{U_{i-1}}{h^2}$

$$+ U_i^3 - f(ih) = 0.$$

Jacobian $J_{ij} = \frac{\partial N_i}{\partial U_j} =$

$$\begin{cases} 0 & \text{if } j \neq i, i+1 \text{ or } i-1. \\ -\frac{1}{h^2} & \text{if } j = i-1 \text{ or } i+1 \\ \frac{2}{h^2} + 1 + 3U_i^2 & \text{if } j = i. \end{cases}$$

---

alternate form:

$$\mathbb{J} = -D_2 + \mathbb{I} + \text{diag}\left(3U_i^2\right).$$

A4. The second order approximation of the fourth derivative on a uniform-grid with spacing h is given by

$$D_4 U_j = \frac{U_{j-2} - 4U_{j-1} + 6U_j - 4U_{j+1} + U_{j+2}}{h^4}$$

Show that the eigenvalues of D_4 on periodic problems are non-negative. *Hints:* use von Neumann analysis and the identity $\cos(2\theta) = 2\cos^2\theta - 1$.

Consider $\underline{U}$ with $U_j = e^{2\pi i j\alpha/N}$.

$$D_4 U_j = \frac{1}{h^4}\left(e^{2\pi i(j-2)\alpha/N} - 4e^{2\pi i(j-1)\alpha/N} + 6e^{2\pi i j\alpha/N} - 4e^{2\pi i(j+1)\alpha/N} + e^{2\pi i(j+2)\alpha/N}\right)$$

$$= \frac{e^{2\pi i j\alpha/N}}{h^4}\left(e^{-4\pi i\alpha/N} - 4e^{-2\pi i\alpha/N} + 6 - 4e^{2\pi i\alpha/N} + e^{4\pi i\alpha/N}\right)$$

$$= \frac{e^{2\pi i j\alpha/N}}{h^4}\left(2\cos 2\theta - 8\cos\theta + 6\right) \qquad \theta = 2\pi\alpha/N.$$

Thus eigenvalues

$$\lambda_\alpha = \frac{1}{h^4}(2\cos 2\theta - 8\cos\theta + 6)$$

$$= \frac{1}{h^4}(4\cos^2\theta - 8\cos\theta + 4) = \frac{4}{h^4}(\cos\theta - 1)^2 \geq 0$$

---

alternate solution: Notice $D_4 = D_2 D_2$, so

$$\lambda_\alpha = \left[\frac{2}{h^2}(\cos\theta - 1)\right]^2 \geq 0.$$

↑ eigenvalues of D_2

Part B: do either B1 or B2

B1. Consider the nonstandard quadrature rule

Correction $-x_2$.

$$\int_{-1}^{1} f(x)dx \approx w_1 f(x_1) + w_1 f(-x_1) + w_2 f'(x_2) - w_2 f'(x_2).$$

It is nonstandard since in involves derivative values. Find relationships between x_1, x_2, w_1 and w_2 so that the quadrature is sixth order accurate.

B2. Consider standard quadratic interpolation on the reference interval $y \in (-1, 1)$:

$$f(y) \approx Q(y) := f(-1)(y^2 - y)/2 + f(0)(1 - y^2) + f(1)(y^2 + y)/2$$

(a) [2 marks] Show with an example that even if $f(-1)$, $f(0)$, and $f(1)$ are all positive, $Q(y)$ can be negative for some values of y. (The right picture would be enough here).

(b) [3] How could you modify the interpolation above so that if $f(-1)$, $f(0)$, and $f(1)$ were all positive, $Q(y)$ would be positive for all y, while maintaining the accuracy of quadratic interpolation? *Hint:* what function always has positive values?

(B1) Note that if $f(x)$ is an odd function, f' is even, $\int_{-1}^{1} f(x)dx = 0$ and the quadrature is exact. Thus, we need only satisfy exactly

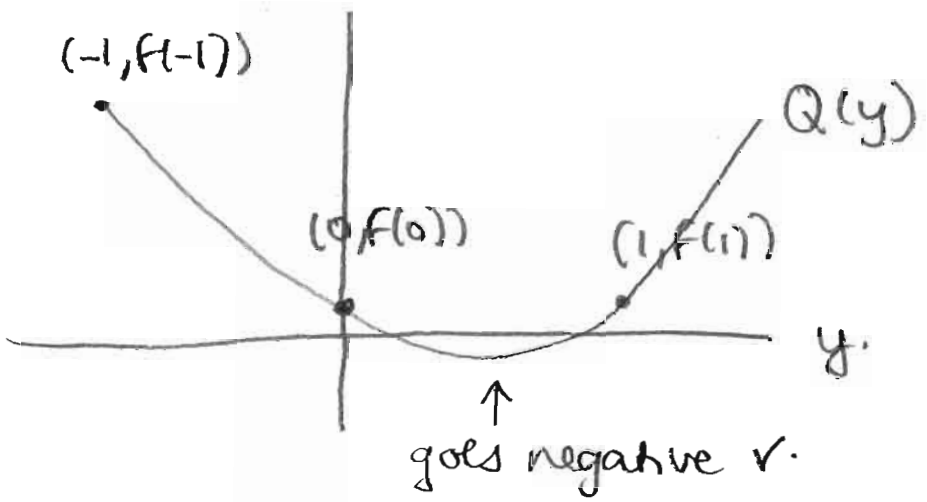
$$\int_{-1}^{1} 1\, dx = 2 \Rightarrow \quad 2w_1 = 2 \Rightarrow w_1 = 1$$

$$\int_{-1}^{1} x^2 dx = 2/3 \Rightarrow \quad x_1^2 + 2w_2 x_2 = 1/3.$$

$$\int_{-1}^{1} x^4 dx = 2/5 \Rightarrow \quad x_1^4 + 4w_2 x_2^3 = 1/5$$

To get a sixth order quadrature.

(B2) a)



b) Do interpolation on $\ln f$:

$$f \approx e^{P(y)}$$

where $P(y) := \ln(f(-1))\,(y^2-y)/2$
$+ \ln(f(0))\,(1-y^2)$
$+ \ln(f(1))\,(y^2+y)/2$

↑
ln values all well defined if f values are positive.

Math 405/607 Review for Exam.

Format: 12 questions, each worth 5 marks, total 60 marks.

11 part A questions
(first two will be short part questions)
1 part B question (choice of 3).

General ideas of discretization, iteration & convergence (two types)

FP accuracy, matrix norms, condition number.

Bisection, Newton, vector Newton methods.

Interpolation and accuracy. Reference Interval
(Taylor series, "g" function arguments)

Quadrature & accuracy, adaptive methods.

Finite difference approximation of derivatives

Finite difference methods for ODE BVPs:
stability & consistency, Lax Equivalence Theorem, implementing boundary conditions, von Neumann analysis

Collocation methods for ODE BVPs:
fourth order (bvp4c) & Spectral.

Time stepping methods:
local error & truncation error.
Stability regions - A- & L-stability
Implicit versus explicit
Stiff problems.

one-step & multi-step methods.
adaptive methods.

Time dependent problems:
heat equation, one-way wave equation, wave equation.
Method of lines.
Stability restrictions
Specialized (non-MOL) time-stepping schemes for undamped oscillators & wave equations.

Poisson Problem in 2D
Finite Difference approx.
Iterative linear algebra:
damped Jacobi & Multi-grid, (on 1D model problem).
[smoothing, restriction, prolongation, coarse grid correction].

Finite Element Methods.
weak form of the problem.
$U(\underline{x})$ in a finite dimensional subspace with basis $\{\psi_j(\underline{x})\}$.
$U(\underline{x})$ satisfies the weak form with test functions $\{\psi_i(x)\}$.
Stiffness & mass matrices - assembly with quadrature methods on reference intervals (triangles or squares in 2D).

The University of British Columbia

Final Examination - Dec 9, 2014

Mathematics 405/607

All Sections

Closed book examination. No calculators. Time: 2.5 hours

Special Instructions:

No books, notes, or calculators are allowed. Show all your work, little or no credit will be given for a numerical answer without the correct accompanying work. Write your answers in booklets provided.

Rules governing examinations

- Each candidate must be prepared to produce, upon request, a UBCcard for identification.
- Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
- No candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination.
- Candidates suspected of any of the following, or similar, dishonest practises shall be immediately dismissed from the examination and shall be liable to disciplinary action.

  (a) Having at the place of writing any books, papers or memoranda, calculators, computers, sound or image players/recorders/transmitters (including telephones), or other memory aid devices, other than those authorized by the examiners.

  (b) Speaking or communicating with other candidates.

  (c) Purposely exposing written papers to the view of other candidates or imaging devices. The plea of accident or forgetfulness shall not be received.
- Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.
- Candidates must follow any additional examination rules or directions communicated by the instructor or invigilator.

Part A. Do all 9 questions. 5 marks each.

A1: Answer the following questions with a brief (one or two sentence) explanation of your reasoning:

(a) [3 marks] A numerical method for ODE boundary value problems is tested with a known solution using uniform step sizes h. The errors for several values of h are given below. What is the convergence order of the method?

h	Error
0.1	0.123
0.05	0.0156
0.025	0.00197

(b) [2] The solution of a nonlinear system is being found iteratively using Newton's method. A test problem with an exact solution is used to investigate the method. Errors after several iterations are shown in the table below. Is the method working properly?

iteration	Error
4	0.895
5	0.448
6	0.224

A2: Consider the second order finite difference approximation of the second derivative:

$$D_2 U_j := \frac{1}{h^2}(U_{j+1} - 2U_j + U_{j-1})$$

where U_j are N values on a grid of the unit interval with equal spacing h ($h = 1/N$). Use von Neumann analysis to find the eigenvalues of D_2 when U_j values are taken to be N-periodic. *Note:* You may have memorized this result, but show details of the derivation for marks on this question.

A3: Consider a reaction diffusion problem for $u(x,t)$ with u 1-periodic in x. It is discretized with a finite difference method in space with spatial steps h and with backward Euler method in time with steps k:

$$U_j^{n+1} = U_j^n + kD_2 U_j^{n+1} + kf(U_j^{n+1})$$

where $U_j^n \approx u(jh, nk)$ and $f(u)$ is a given function. $\mathbf{U}^{n+1}$ is vector with N components ($N = 1/h$). At every time step, the equation above is a nonlinear system for $\mathbf{U}^{n+1}$. Write out the entries of the Jacobian matrix for this system. *Note:* some entries will involve values of f'.

The questions **A4**, **A5** and **A6** below concern the following approximation of 1-periodic in space solutions $u(x,t)$ of the one-way wave equation $u_t + u_x = 0$:

$$U_j^{n+1} = U_j^n - \frac{k}{h}(U_j^n - U_{j-1}^n).$$

Initial data $\mathbf{U}^0$ are given. Time steps are taken proportional to space steps:

$$k = Ch$$

with $C = 0.9$.

A4: Show that the scheme satisfies a maximum principle, that is that

$$\max_j U_j^n \leq \max_j U_j^0$$

for every $n > 0$.

A5: What is the order of the truncation error of the scheme?

A6: Consider the modified scheme

$$U_j^{n+1} = U_j^n - \frac{k}{h}(U_{j+1}^n - U_j^n).$$

with the same initial data and time steps. Use von Neumann analysis to show that this scheme is not stable.

A7: Consider the following time stepping method for the problem $\dot{u} = f(u,t)$:

$$U^{n+1} = U^n + \frac{k}{2}\left(3f(U^n, nk) - f(U^{n-1}, (n-1)k)\right)$$

Identify the order of the truncation error of the method. Other aspects of this method are considered in problem B1.

A8: Consider approximating the function $f(x)$ on the interval [-1,1] using a linear function $L(x) = Ax + B$ that agrees with f at two distinct points x_1 and x_2 in the interval, that is $L(x_1) = f(x_1)$ and $L(x_2) = f(x_2)$. Show that

$$|L(x) - f(x)| \leq CK_2$$

where

$$K_2 = \max_{x\in[-1,1]} |f''|$$

and C is a constant that depends on the choice of x_1 and x_2. Other aspects of this approximation are considered in problem B2.

A9: Consider the following problem for scalar functions $x(t)$ and $y(t)$:

$$\frac{dx}{dt} = f(x,y) \quad (1)$$
$$g(x,y) = 0 \quad (2)$$

where the functions f and g are given. Initial values $x(0)$ and $y(0)$ are also given, that satisfy equation (2).

(a) [3 marks] Describe a consistent numerical method for time stepping approximate solutions $X^n \approx x(nk)$ and $Y^n \approx y(nk)$, where k is given time step. It is desirable that (2) be satisfied by the approximation, that is

$$g(X^n, Y^n) = 0$$

to high accuracy (machine precision) at each n.

(b) [2] Describe how you would test your method above. Give enough detail that someone could implement your test.

Part B. Do *one* of the following three problems. 5 marks

B1: Consider the time stepping method from A7, for approximating $\dot{u} = f(u,t)$:

$$U^{n+1} = U^n + \frac{k}{2}\left(3f(U^n, nk) - f(U^{n-1}, (n-1)k)\right).$$

Determine what points (if any) on the negative real axis are in the stability region of the method.

B2: Consider approximating the function $f(x)$ on the interval [-1,1] using a linear function $L(x) = Ax + B$ that agrees with f at two distinct points x_1 and x_2 in the interval as considered in A8. It was shown that

$$|L(x) - f(x)| \leq CK_2$$

where

$$K_2 = \max_{x\in[-1,1]} |f''|$$

and C is a constant that depends on the choice of x_1 and x_2. Identify the values of x_1 and x_2 that make the constant C above as small as possible.

B3: Consider the following equation for $u(x,t)$ defined on the real line $x \in (-\infty, \infty)$:

$$u_{tt} - u_t = u_{xx} + f(x,t)$$

where the functions $f(x,t)$, $u(x,0)$ and $u_t(x,0)$ are given. These functions are zero outside the x-interval [-1,1]. The solution satisfies (for each t):

$$\lim_{|x|\to\infty} u(x,t) = 0$$

(a) [3 marks] Describe a numerical scheme to approximate the solution of this problem. Include details on how you will handle the infinite domain aspect of it.

(b) [2] Consider your scheme in the periodic domain setting. Use von Neumann analysis to determine the time step restrictions your method will have (if any).