INTRODUCTION TO SCIENTIFIC COMPUTING PART II –DRAFT

July, 2001

These are notes and those of math475A are used in the upper level undergraduate courses on numerical analysis. Since the students taking this course sequence come from diverse backgrounds and most of them do not know any analysis, we have focused on scientific computing, rather than numerical analysis.

The notes are being extensively revised. The major revisions are: (1) substantial correction of errors. (2) Incorporation of many more examples (and removal of examples that, although attributed to the original author, are not original). (3) Restructuring of the course material. (4) Making greater use of hyperlinks in order to reduce the complexity of the notes while at the same time making the cross references a useful feature of the text. (5) Changing the notes to PDF format. These notes were never intended to be public, but the wonders of the web have made this academic now. This is a *work in progress*, and as such, it is bound to have many errors, primarily of typographic nature. We would be delighted to hear from you, if these notes have been of any use to you. We are particularly interested in receiving corrections and suggestions.

At some point in time the use of matlab in numerical analysis classes was quite rare. Then, making a statement like we only use matlab in this numerical analysis sequence, was considered bold...fortunately, no longer. We prevailed and now the reasons are perfectly obvious. If you happen not to use or know how to use matlab, don't fret: just go to the home page and download the matlab primer. Just read carefully the first 20 pages. Then, do homework 1 of 475A. This will teach you enough of the essentials of matlab to get going.

Please note that the algorithms given in these notes are not matlab compatible with regard to indices. Be careful, if you are new to matlab, to adjust the algorithms for indices that include non-positive values.

The sequence followed in these notes results from two things: The assumption that matlab might be new to the student. Since we go from scalar, to vector, to matrix problems, the student will have time to learn enough matlab so that by the time matrix problems come around, they will

be proficient. The more important reason, however, is that it is important to emphasize the notion of norm and of error. I found it better to work up in complexity, from scalar, to vector, to matrix norms.

The original set of notes was used in teaching the year sequence in numerical analysis at the senior undergraduate level. The notes became useful to our applied math graduate students, taking their first year of numerical analysis. I had help in writing/editing of these notes by Prof. Rob Indik, Ms. Emily Lane, and Ms. Rachel Labes; their input and their hard work has yielded a better set of notes. Whatever errors are still in the notes are all my own ....Juan Restrepo

Contents

Part I

ORDINARY DIFFERENTIAL EQUATIONS

0.1 The INITIAL VALUE PROBLEM (IVP)

ORDINARY DIFFERENTIAL EQUATIONS (ODE) Material for this section is taken from books by Kincaid & Cheney, Burden & Faires, Atkinson, Iserles, Isaacson & Keller, Coddington & Levinson, Stoer & Burlisch. The references to numerical analysis books can be found by clicking here.

The first part will consider the "initial value problem." (IVP) in detail. The second part will present the "boundary value problem" (BVP) in a cursory way (see notes for 575B course, where we use variational methods to recast the BVP for its numerical solution, a powerful and elegant analytical technique that leads to a host of important numerical schemes).

An ordinary differential equation is a function that maps $x \mapsto y(x) \in R^n$, n a natural number, and it involves x the independent variable, $y(x)$, and finite set of derivatives of $y(x)$. It has the form

$$g\left(x, y, y', y'', \ldots, \frac{d^m y}{dx^m}\right) = 0. \tag{1}$$

where $g \in R^k$, k a natural number. In many instances $k = 1$. In fact, assume for now assume that $k = 1$ in what follows. Equation (1) is an m^{th} order ODE, since the highest non-zero derivative in y is $\dfrac{d^m y}{dx^m}$.

We can recast (1) in "normal form." Solving for $\frac{d^m y}{dx^m}$ we have

$$\frac{d^m y}{dx^m} = f\left(x, y, y', \ldots, \frac{d^{m-1}y}{dx^{m-1}}\right).$$

Let $\boxed{y^{(j)} = \dfrac{d^j y}{dx^j}}$ with $j = 0, 1, 2, \ldots m$ with $y^{(0)} \equiv y$.

Then (1) is equivalent to $\dfrac{d}{dx}\begin{pmatrix} y^{(0)} \\ y^{(1)} \\ : \\ y^{(m-1)} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ : \\ f(x, y^{(0)}, y^{(1)} \cdots y^{(m-1)}) \end{pmatrix}$,

or $\dfrac{dY}{dx} = F(x, Y)$
with $Y = (y^{(0)}, \cdots, y^{(m-1)})^T$
and $F \equiv (f^{(0)}, f^(1), \ldots, f^{(m-1)})^T$,

where the superscript T stands for transpose.

Definition: Autonomous and Non-autonomous ODE'S:

(2)
$$\frac{dY}{dx} = F(Y) \quad \text{is autonomous since } x \text{ does not appear explicitly.}$$
$$\frac{dY}{dx} = F(Y, x) \quad \text{is non-autonomous.}$$

Non-autonomous ODE's can be recast as an ODE by the following procedure:

let $Y = \left(y^{(0)}, \cdots, y^{(m)}\right)^T$. It's an $m+1$ dimensional vector

$$F = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ : \\ f(y^{(0)}, \cdots, y^{(m)}) \\ 1 \end{pmatrix}, \text{is an } m+1 \text{ dimensional vector,}$$

where

$$y^{(m)} = x$$
$$\frac{dy^{(m)}}{dx} = 1 \text{ becomes the last equation of (2)}$$

The Initial Value Problem is defined as:

$$\begin{cases} \dfrac{dY}{dx} = F(Y, x) \\ \\ Y(x_0) = Y_0 \qquad Y_0 \text{ is an } m \text{ dimensional vector.} \end{cases}$$

Example) We recast the following initial value problem as a normalized autonomous system.

$$\sin t y''' + \cos(ty) + \sin(t^2 + y'') + (y')^3 = \log t$$
$$y(2) = 7$$
$$y'(2) = 3$$
$$y''(2) = -4$$

Let $\quad y^{(0)} = y, \quad y^{(1)} = \dfrac{dy^{(0)}}{dt}, \quad y^{(2)} = \dfrac{dy^{(1)}}{dt}.$

Normalizing:

$$y''' = -\frac{1}{\sin t}\left(\cos(ty) + \sin(t^2 + y^{(2)}) + (y^{(1)})^3\right) + \frac{\log t}{\sin t}$$

$$\begin{cases} \frac{d}{dt}y^{(2)} = -\frac{1}{\sin t}(\cos(ty^{(0)}) + \sin(t^2 + y^{(2)}) + (y^{(1)})^3 + \frac{\log t}{\sin t} \\ \frac{d}{dt}y^{(1)} = y^{(2)} \\ \frac{d}{dt}y^{(0)} = y^{(1)} \end{cases}$$

Let $Y = (y^{(2)}, y^{(1)}, y^{(0)})^T$

then $\frac{dY}{dt} = F(t, Y) \equiv (f^{(2)}, f^{(1)}, f^{(0)})^T$

with $F = \begin{cases} -\frac{1}{\sin t}(\cos(ty^{(0)}) + \sin(t^2 + y^{(2)}) + (y^{(1)^3}) + \frac{\log t}{\sin t} \\ y^{(2)} \\ y^{(1)} \end{cases}$

$\underline{Y(2) = (-4, 3, 7)^T}$

Made autonomous: $\frac{dY}{dx} = f(Y) = (f^{(3)}, f^{(2)}, f^{(1)}, f^{(0)})$

where $f^{(3)} = 1$

and $Y = (y^{(3)}, y^{(2)}, y^{(1)}, y^{(0)}), \quad y^{(3)} = t \quad Y(2) = (2, -4, 3, 7)$

□

0.1.1 Some important theorems on ODE's

see Braun, for an introductory exposition, and Coddington & Levinson and Birkhoff & Rota for a more advanced one

Definition:

A function $f(x, y)$ satisfies a "Lipschitz Condition" in the variable y on a set $S \in \mathbb{R}^2$ provided $\exists$ constant L such that

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$$

whenever $(x, y_1), (x, y_2) \in S$. L is the Lipschitz constant.

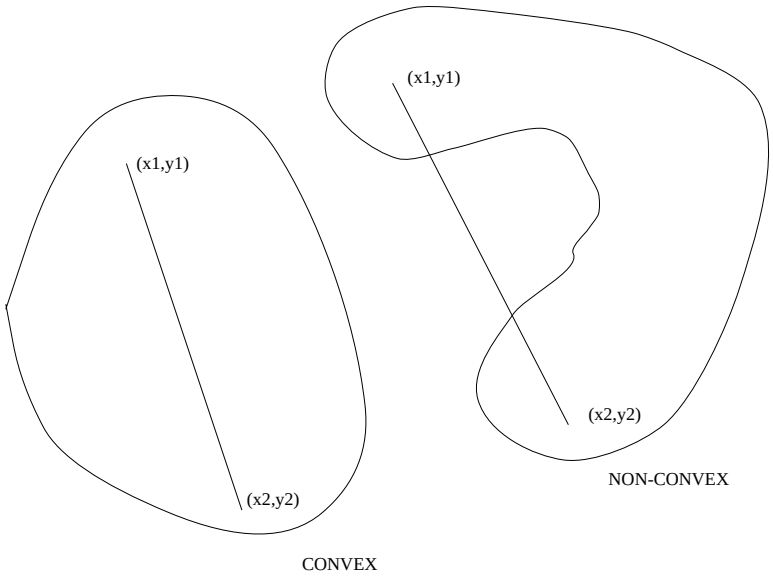

Figure 1: CONVEX AND NON-CONVEX EXAMPLES

Definition: A set $S \in \mathbb{R}^2$ is convex, if whenever $(x_1, y_1)\,\&\,(x_2, y_2)$ belong to S, the point $[(1-\lambda)x_1 + \lambda x_2, (1-\lambda)y_1 + \lambda y_2]$ also belongs to S for each λ when $0 \leq \lambda \leq 1$. See Figure 1.

We study the IVP

$$(3) \qquad \begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases} \quad \text{a system of } n \text{ ODE's}$$

$$\text{Take } S \equiv \{(x,y) | a \leq x \leq b, y \in \mathbb{R}^n\}$$
$$\text{with } a, b, \text{ finite}, a \leq x_0, \leq b.$$

It has exactly 1 solution provided f satisfies the following conditions of existence and uniqueness.

Existence & Uniqueness

Theorem. f defined and continuous on S, convex (see 0.1.1), and satisfies a Lipschitz condition 0.1.1 $\Rightarrow$ for every $x_0 \in [a,b]$ & every $y_0 \in \mathbb{R}^n$ there exists 1 function $y(x)$ such that

a) y is continuous and differentiable for $x \in [a,b]$

b) $y'(x) = f(x, y(x))$ for $x \in [a, b]$

c) $y(x_0) = y_0$

□

Theorem. Suppose f defined and convex. If there exist a constant $L > 0$ such that

$$\left| \frac{df_i}{dy_j} \right| \leq L \quad \forall (x, y) \in S$$

$$\Rightarrow f \text{ satisfies a Lipschitz condition on } S$$

□

Remark. In applications one finds that f is usually continuous in S and also continuously differentiable there, but could have the derivates $\frac{df_i}{dy_j}$ unbounded on S.
$\Rightarrow$ While (3) is still solvable the solution may only be defined in some $U(x_0)$ neighborhood of the initial $x_0 \in [a, b]$.

Example)

$$\begin{cases} y' = y^2 \\ y(0) = 1 \end{cases}$$

has solution $y = \frac{1}{1 - x}$ defined only for $x < 1$.

Continuous Dependence:

Theorem. Let $f : S \to \mathbb{R}^n$ continuous on S also satisfying the Lipschitz condition 0.1.1

$||f(x, y_1) - f(x, y_2)|| \leq L||y_1 - y_2||$

$\forall (x, y) \in S, \quad i = 1, 2.$ Let $a \leq x_0 \leq b$. For the solution $y(x; s)$ of the initial value problem

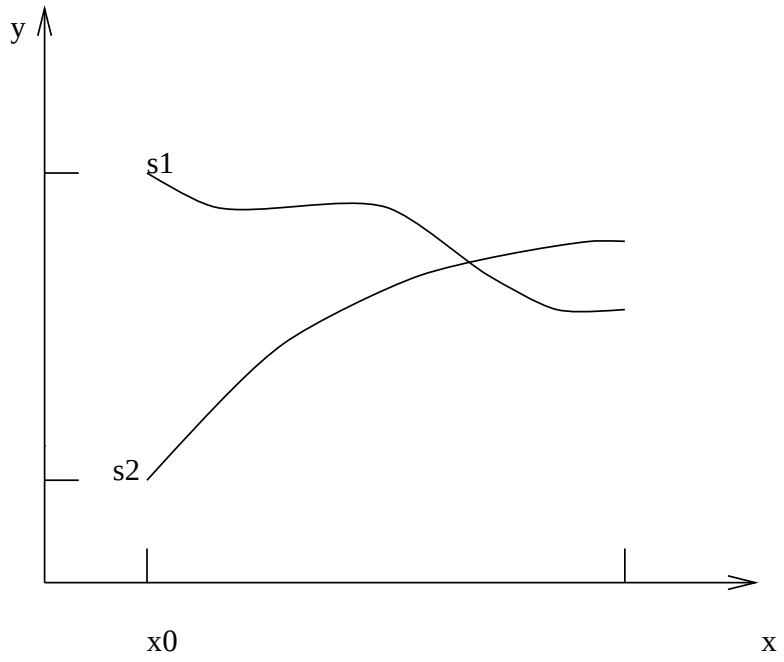

Figure 2: Trajectories y_1 and y_2 emanating from initial data s_1 and s_2, respectively

$$\begin{cases} y' = f(x,y) \\ y(x_0;s) = s \end{cases}$$

there holds the estimate

$$||y(x;s_1) - y(x;s_2)|| \leq e^{L|x-x_0||}|s_1 - s_2||$$

for $a \leq x \leq b$.

Proof. See Figure 0.1.1

$$y(x;s) = s + \int_{x_0}^{x} f(t, y(t,s))dt, \qquad \text{then}$$
$$y(x;s_1) - y(x_i;s_2) = s_1 - s_2 + \int_{x_0}^{x} [f(t, y(t,s_1)) - f(t, y(t_1,s_2))]dt$$

thus

$$(4)\quad ||y(x;s_1) - y(x;s_2)|| \leq ||s_1 - s_2|| + L| \underbrace{\int_{x_0}^{x} ||y(t;s_1) - y(t;s_2)||dt|}_{\Phi(x)}$$

then $\Phi'(x) = ||y(t, s_1) - y(t, s_2)||$. Thus by (4), for $x \geq x_0$
$\alpha(x) \leq ||s_1 - s_2||$ with $\alpha \equiv \Phi'(x) - L\Phi(x)$

Take the initial value problem

$$\begin{cases} \Phi'(x) = \alpha(x) + L\Phi(x) \\ \Phi(x_0) = 0 \end{cases}$$

for $x \geq x_0 \Rightarrow \Phi(x) = e^{L(x-x_0)} \int_{x_0}^{x} \alpha(t) e^{-L(t-x_0)} dt$.

Since $\alpha \leq ||s_1 - s_2||$, then

$$\begin{aligned} 0 \leq ||\Phi|| &\leq e^{L(x-x_0)} ||s_1 - s_2|| \int_{x_0}^{x} e^{-L(t-x_0)} dt \\ &= \frac{1}{L} ||s_1 - s_2|| [e^{L(x-x_0)} - 1] \quad x \geq x_0 \end{aligned}$$

Since $\alpha = \Phi' - L\Phi \Rightarrow ||y(x, s_1) - y(x, s_2)|| = \Phi'(x) = \alpha(x) + L\Phi(x) \leq ||s_1 - s_2|| e^{L|x-x_0|}$ □

The above theorem can be sharpened: under extra continuity, the solution of the IVP actually depends on initial value in a continuously differentiable manner:

Theorem. If, in addition to assumptions in previous theorem and if the Jacobian $D_y f(x, y) \equiv [\partial f_i / \partial y_j]$ exists on S, and is continuous and bounded,

$$||D_y f(x, y)|| \leq L \text{ for} (x, y) \in S,$$

$\Rightarrow$ the solution $y(x, s)$ of $y' = f(x, y), y(x_0, s) = s$ is continuously differentiable for all $x \in [x_0, b]$ and all $s \in \mathbb{R}^n$

□

Example)

$$\begin{aligned} y' &= 1 + \sin(xy) = f(x, y) \\ S &= \{(x, y) | 0 \leq x \leq 1, -\infty < y < \infty\} \\ \frac{\partial f}{\partial y} &= x \cos(xy) \Rightarrow L = 1 \end{aligned}$$

$\therefore$ for any (x_0, y_0) with $0 < x_0 < 1 \exists$ a $Y(x)$ and associated IVP on some interval $[x_0 - \alpha, x_0 + \alpha] \in [0, 1]$.

Example) $y' = \dfrac{2x}{a^2} y^2 \qquad y(0) = 1 \qquad a > 0$ const.

$Y(x) = \dfrac{a^2}{a^2 - x^2} \quad -a < x < a$. Solution depends on size of a.

To determine L take $\frac{\partial f}{\partial y}(x, y) = \frac{4xy}{a^2} \therefore$ to have L finite on S, S must be bounded in x and y, say $-c \leq x \leq c, -b \leq y \leq b$.

Theorem. (Existence)

(5) $$\text{For } \begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

if f is continuous in a rectangle S with center at (x_0, y_0), say,

$$S = \{(x, y) : |x - x_0| \leq \alpha, |y - y_0| \leq \beta\}$$

then the initial value problem (5) has a solution $y(x)$ for $|x - x_0| \leq \min(\alpha, \beta/M)$, where M is the maximum of $|f(x, y)|$ in S.

Example)

$$y' = (x + \sin y)^2$$
$$y(0) = 3$$
$$\text{has a solution on } -1 \leq x \leq 1$$

$f(x) = (x + \sin y)^2$ and $(x_0, y_0) = (0, 3)$

$$S = \{(x, y) : |x| \leq \alpha, |y - 3| \leq \beta\}$$

If $(x, y)| \leq (\alpha + 1)^2 \equiv M$. Want $\min(\alpha, \beta/M) \geq 1$ so set $\alpha = 1$. Then $M = 4$ and everything is consistent if $\beta \geq 4$. So theorem asserts that a solution exists on $|x| \leq \min(\alpha, \beta/M) = 1$.

□

A useful theorem: Consider

(6) $$\begin{cases} Y' = T(x)Y \\ Y(a) = I \end{cases}$$

where $T(x)$ is an $n \times n$ matrix

Theorem. If $T(x)$ is continuous on $[a, b]$ and $k(x) \equiv ||T(x)|| \Rightarrow$ solution $Y(x)$ of (6) satisfies

$$||Y(x) - I|| \leq e^{\int_a^x k(t)dt} - 1 \quad x \geq a$$

Proof: exercise. □

0.1.2 Numerical Methods for the approximate solution of ODE'S.

We concentrate first on the IVP and then discuss the BVP. We only consider finite difference methods (F.D.). F.D. methods invariably require that the independent variable x be a discrete sequence and that $Y(x)$, derivatives of $Y(x)$, and coefficients in the equation that depend on x be approximated on such a grid.

Notation let $Y(x)$ be the true solution of

$$\begin{cases} Y' = f(x, Y) \\ Y(a) = Y_0. \end{cases} \tag{7}$$

let $y(x)$ represent the approximate solution and by way of notation, let

$$y(x_0) \equiv y_0, y(x_1) \equiv y_1, \ldots y(x_n) \equiv y_n. \tag{8}$$

let y_h denote an approximation at some resolution, given by h the grid spacing. If the grid is equally spaced

$$x_n = x_0 + nh, n = 0, 1, 2 \cdots \tag{9}$$

Take $x_0 = a$, for simplicity, and let $N(h)$ denote the largest index N for which

$$x_N \leq b \qquad x_{N+1} > b$$

where $a < b$. The simplest finite difference approximation would be

$$\begin{cases} y(x_{n+1}) = y_n + h_n f(x_n, y_n) \\ y_0 \cong Y_0, \end{cases} \quad \text{where } h_n = x_{n+1} - x_n, \text{ and } n = 0, 1 \ldots.$$

Using an equally spaced grid the above scheme would be Euler(Forward Euler)

Perhaps the simplest most straightforward scheme. It reads

$$y_{n+1} = y_n + hf(x_n, y_n) \qquad n = 0, 1, \ldots \tag{10}$$
$$y_0 \cong Y_0$$

where y_n is consistent with (8) and x_n as per (9). Following Atkinson's suggestion, it is very useful to interpret the Foward Euler scheme in a variety of ways. Look at the interpretation of taking a single step:

0.1.3 Generalizations of Forward Euler by its Different Interpretations

1. Geometric Interpretation. See Figure 0.1.3.

$\frac{D\,y}{h} = Y'(x_0) = f(x_0, Y_0)$

$Y(x_1) - Y(x_0) \approx D\,y = hY'(x_0) \Rightarrow Y(x_1) \approx Y(x_0) + hf(x_0, Y(x_0))$,

Repeating the argument for $[x_1, x_2], [x_2, x_3] \cdots$

2. Taylor Series

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \underbrace{\frac{h^2}{2}Y''(\xi_n)}_{T_n = \frac{h^2}{2}Y''(\xi_n)}$$

3. Numerical Differentiation

$\frac{Y(x_{n+1}) - Y(x_n)}{h} \approx Y'(x_n) = f(x_n, Y(x_n))$

$\therefore Y(x_{n+1}) \cong Y'(x_n) + hf(x_n, Y(x_n))$

4. Numerical Integration. See Figure 0.1.3

Integrate $Y'(x) = f(x, y)$

Over $[x_n, x_{n+1}]$

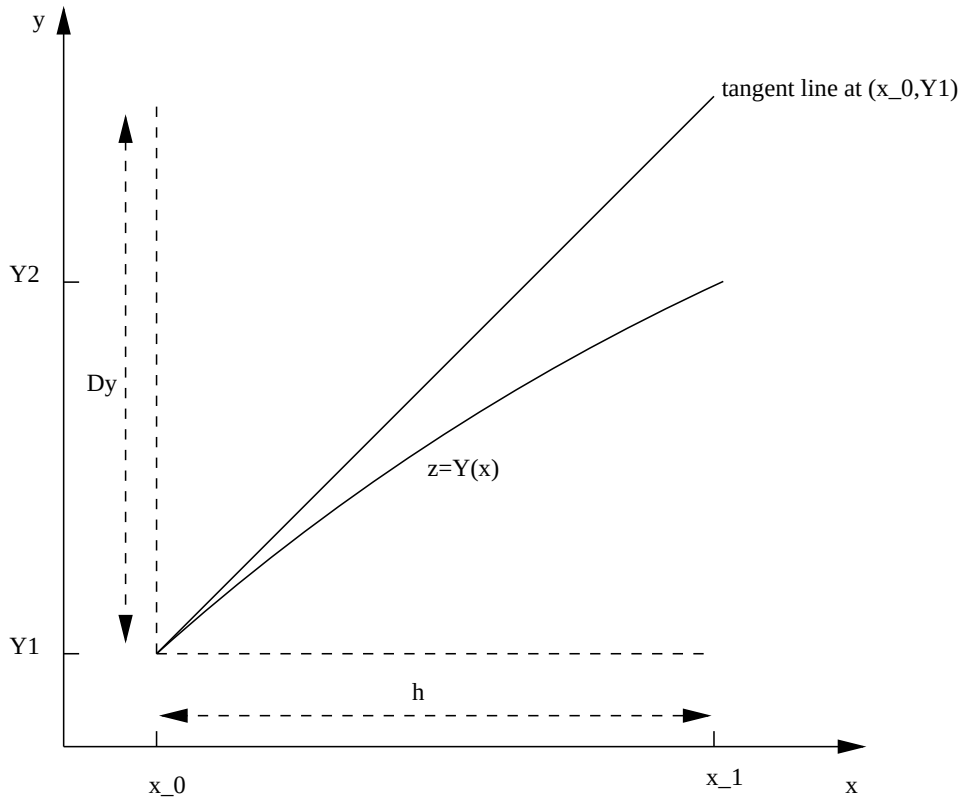

Figure 3: Geometrical interpretation of a single-step using Forward Euler scheme

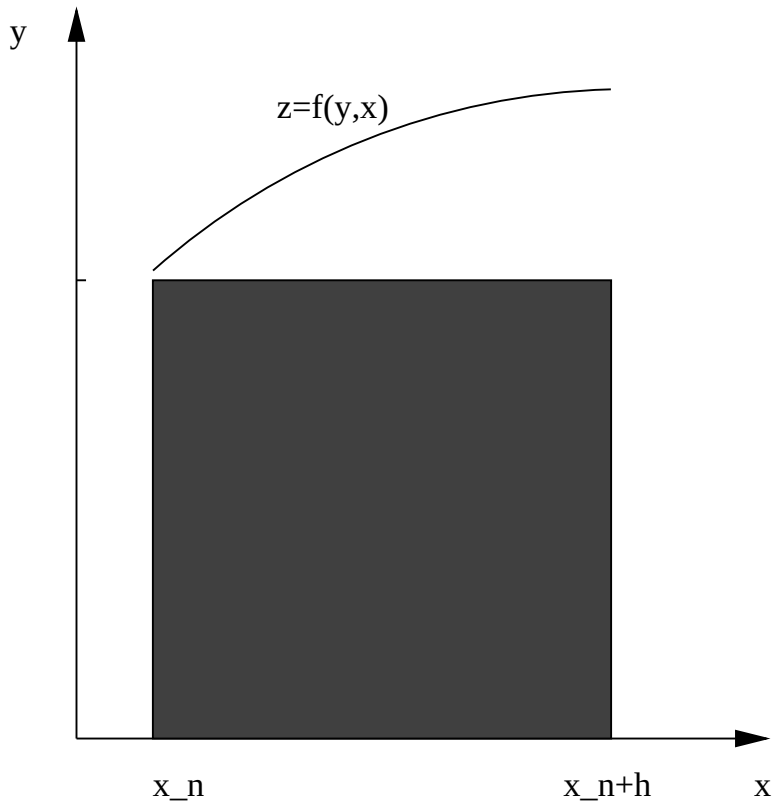

Figure 4: Quadrature interpretation of single-step using Forward Euler

$$Y_{n+1} = Y_n + \underbrace{\int_{x_n}^{x_{n+1}} f(t, Y(t))dt}_{\text{L.H.S} \approx hf(x_n, Y_n)}$$

Remark. Interpretations (2) and (4) above form the basis of a set of methods that are progressively more accurate.

$$\begin{cases} \text{(2) generalizes to what are known as SINGLE STEP} & \text{METHODS} \\ \text{(4) generalizes to what are known at MULTI STEP} & \text{METHODS.} \end{cases}$$

Interpretation (3) $\rightarrow$ doesn't lead to many possibilities, but leads to a way to solve stiff equations (to be discussed later). It also leads to an academically interesting case, the Midpoint method, which is ideal to introduce the concept of instability in the context of the approximate solution of ODE's.

□

0.1.4 Errors in the Numerical Approximation of the IVP

The numerical solution of (10) is an approximation to the solution of (7), provided certain conditions are met and will be discussed presently. Why is the numerical solution called an approximation? Because invariably there are errors made. The types of errors incurred in the approximation are global and local "truncation" and "round-off" errors, along with errors related to to machine representation. In what follows we will group round-off and machine representation of numbers in one group and truncation errors in another group. The truncation errors are present regardless of exactness of machine representation of numbers or how the computation on the numerical scheme is carried out. Roundoff error and machine errors have to do with how the computation is carried out and on what type of machine. The Total Error is the accumulation of both types of error.

Local (and Global) Truncation = error made in one step when we replace an infinite process by a finite one (independent of round off error). (Global is sum over all steps). The Local Round off = error made by computing with limited precision on one step. (Global is cummulative round-off error). Total Error = sum of roundoff and truncation error. The sources of roundoff

error are no different than those considered in the previous semester. The new ones, which we will devote our attention, is the truncation error. We'll mostly ignore rounding errors for now.

Example Suppose we use forward Euler to approximate the solution of

$$\begin{cases} Y' = f(x, Y) \\ Y(a) = Y_0 \end{cases}$$

and compare y_h to true solution $Y(x)$ as follows: make a table (assume for simplicity that stepsize is constant and of size h). The table contains $y_h(x_n)$ for different values of h.

The table and its analysis constitute a "convergence analysis." Most likely, we don't have $Y(x)$, the exact answer. Suppose we do.

				"error" $\equiv E_h(x)$
h	x	$y_n(x)$	$Y(x)$	$\lvert y_n - Y_(x)\rvert$
0.20	0.40	approx	true	:
	0.80	:	:	:
	1.20			:
	1.60			:
	:			:
0.10	1.40	approx	true	:
	0.80	:	:	:
	1.20			:
	1.60			:
	:			:
0.05	1.40	approx	true	:
	0.80	:	:	:
	1.20			:
	1.60			:
	:			:
	etc.			

What you see:

(1) if numerical method is convergent $\Rightarrow$ as $h \to 0$ $\quad y_h(x_n)$ will approach $Y(x_n)$ uniformly.

(2) The last column shows the cummulative effect of errors, if any, in the integration, as a function of x_n for a given h. We will call this the "Global error" $\equiv E_h(x_n)$

Take $E_{0.20}(x^\star), E_{0.10}(x^\star), E_{0.05}(x^\star) \cdots$, corresponding to h being halved, and the absolute difference $|y_h - Y|$ at some $x = x^\star$. For Euler, we'll see that the error will drop by $\dfrac{1}{2}$ if we have. Since error drops proportional to h, we say the method's local truncation error is of order h.

Another way we can convey the global error is by making a plot of $E(h)$ as a function of h. The convention is to plot $E(h)$ on the vertical axis and h on the horizontal axis, with h DECREASING along the right. Furthermore, the plot should be a log-log plot. One picks a location x^*, sufficiently far from x_0 (this is determined largely on common sense). The discretization is picked so that for any x^* is a value taken by x_n for any given h. At this location the error $E(h)$ is recorded as a function of h for the same initial data. The plot will show how this global error behaves as h is changed. Moreover, as we will see later on, the slope of the log plot will indicate the "convergence rate" of the method, if the method converges. More on this later. $\square$

Exercise. In some rare instances one can actually solve the resulting difference equation analytically. For example, we wish to solve

$$\begin{cases} Y' = 2x \\ Y(0) = 0. \end{cases} \tag{11}$$

Verify that the exact solution is $Y(x) = x^2$. $\quad$ Let $y_n \equiv y(x_n), \quad x_n = nh$ $n = 0, 1, \cdots$

The forward Euler approximation to (11) is $y_{n+1} = y_n + 2hx_n$, with $y_0 = 0$.

Solution of difference equation: (see difference equations from previous semester, or use induction)

$$y_n = x_n x_{n-1} \quad n \geq 1$$

$$\therefore E_n \equiv |Y(x_n) - y_n| = |x_n^2 - x_n x_{n-1}| = |hx_n|$$

$\therefore$ Global error for each fixed value x is proportional to h

□

There's more to the error analysis, of course. As always, the goal of scientific computing and numerical work is not to compute exactly, but to know exactly what errors are made. We need to learn a number of very useful theorems, which we can be used to tell whether we can have confidence in the answer obtained in a computation (remember as well that computers ALWAYS give answers), and can be extended to design numerical solutions to your practical problems. These theorems, together with detailed and careful work on the assignments will go a long way to teach you the basics of numerical analysis and scientific computing.

0.1.5 How is the Approximation Related to the IVP, if at all?

The Big Questions about any scheme used to approximate an ODE and its solutions are:

(a) Is Method Convergent?

(b) if so, How fast does error $\searrow 0$ as $h \searrow 0$?

(c) Is method Consistent?

(c) Is method Stable?

(d) Is this the most computationally-efficient method to solve the ODE?

We begin to learn these concepts and also learn how to prove these for a variety of different schemes.

First, a useful lemma:

Lemma. For any real x

$$1 + x \leq e^x$$

for any $x \geq -1$ $\quad 0 \leq (1+x)^m \leq e^{mx} \quad m > 0$

Proof. Taylor $\quad e^x = 1 + x + \underbrace{\frac{x^2}{2} e}_{\text{remainder}}$ always > 0 between 0 and x.

i.e.

$$0 \leq 1 + x \leq 1 + x + \frac{1}{2} x^2 e^{\xi} = e^x.$$

The second statement trivially follows.

□

For simplicity, assume that $f(x, y)$ satisfies stronger Lipschitz condition:

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2| \left\{ \begin{array}{l} -\infty < y_1, y_2 < \infty \\ x_0 \leq x \leq b \end{array} \right.$$

$L > 0$ (this condition is stronger than necessary $\cdots$ it just simplifies proofs and avoids technicalities that can be mastered after this case is understood).

Remark. Forward Euler (see 0.1.2) is not the best ODE integrator: but it is the simplest.

Theorem.

(Convergence for Forward Euler) Assume $Y(x)$ is solution of IVP (with f satisfying the Lipschitz condition) and

$$\begin{aligned} ||Y''(x)||_\infty \leq M \, \forall x \in [x_0, b]. \\ \Rightarrow \{y_h(x_n) | x_0 \leq x_n \leq b\} \\ \text{obtained by Euler method satisfies} \\ \max_{x_0 \leq x_n \leq b} |Y(x_n) - y_h(x_n)| \leq e^{(b-x_0)L} |e_0| \\ + \frac{e^{(b-x_0)L} - 1}{L} \tau(h) \end{aligned}$$

where $\tau(h) = \dfrac{h}{2}M$ and L is the Lipschitz constant associated with the IVP and

$$e_0 = Y_0 - y_h(x_0).$$

If, in addition, $|Y_0 - y_h(x_0)| \le c_1 h \quad$ as $h \to 0$

for some $c_1 \ge 0$ (e.g. $Y_0 = y_0$ for all h) $\Rightarrow \exists \quad B \ge 0$ constant

$$\max_{x_0 \le x_n \le b} |Y(x_n) - y_h(x_n)| \le Bh$$

Remark. How big is B? Could be large!!

Proof. Let $e_n = Y(x_n) - y(x_n), \quad n \ge 0.$

let $\tau_n = \dfrac{h}{2}Y''(\xi_n) \quad 0 \le n \le N(h) \equiv$ number of steps depends on h for fixed $[x_0, b]$ interval. Also,

$$x_n \le \xi \le x_{n+1}$$

estimate: $\max_{0 \le n \le N-1} |\tau_n| \le \tau(h) = \dfrac{h}{2}M$

Now,

$$\begin{aligned} Y_{n+1} &= Y_n + hf(x_n, Y_n) + h\tau_n \\ y_{n+1} &= y_n + hf(x_n, y_n) \qquad 0 \le n \le N(h) - 1 \end{aligned}$$

subtracting:

$$\begin{aligned} (12) \qquad \therefore \quad & e_{n+1} = e_n + h[f(x_n, Y_n) - f(x_n, y_n)] + h\tau_n \\ & |e_{n+1}| \le |e_n| + hL \underbrace{|Y_n - y_n|}_{e_n} + h|\tau_n| \\ & |e_{n+1}| \le (1 + hL)|e_n| + h\tau(h) \qquad 0 \le n \le N(h) - 1 \end{aligned}$$

Apply (12) recursively:

$$|e_n| \le (1 + hL)^n |e_0| + \{1 + (1 + hL) + \cdots (1 + hL)^{n-1}\} h\tau(h)$$

Recall: $1 + r + r^2 + \cdots r^{n-1} = \dfrac{r^n - 1}{r - 1} \qquad r \neq 1.$

$$\therefore |e_n| \leq (1 + L)^n |e_0| + \Big[\frac{(1 + hL)^n - 1}{L}\Big]\tau(h)$$

using $(1 + hL)^n \leq e^{nhL} = e^{(x_n - x_0)L} \leq e^{(b - x_0)}L$

implies main result:

$\therefore$

$$\max |Y(x_h) - y_n(x_n)| \leq e^{(b-x_0)L}|e_0| + \frac{e^{(b-x_0)L} - 1}{L}\tau(h)$$

The

$$\max |Y(x_n) - y_n(x)| \leq Bh \tag{13}$$

follows trivially from $\nearrow$ above, with

$$B = c_1 e^{(b-x_0)L} + \left[\frac{e^{(b-x_0)L} - 1}{L}\right]\frac{M}{2}$$

since by assumption $|Y_0 - y_h(x_0)| \leq c_1 h \quad$ as $\quad h \to 0$

□

Equation (13) gives rate of convergence of method. (Parenthetically, a method that does not converge is useless). But B estimate may be too large. We can sharpen the estimate if the following holds:

Corollary.

Same hypothesis as previous theorem. In addition

$$\frac{\partial f}{\partial y}(x, y) \leq 0 \left\{ \begin{array}{c} x_0 \leq x \leq b \\ \infty < y < \infty \end{array} \right.$$

$\Rightarrow$ For h sufficiently small

$$|Y(x_n) - y(x_n)| \leq |e_0| + \frac{h}{2}(x_n - x_0) \max |Y''(x)|$$

for $x_0 \leq x_n \leq b$.

Proof. Apply Mean Value Theorem to

$$e_{n+1} = e_n + h[f(x_n, Y_n) - f(x_n, y_n)] + h\tau_n \tag{14}$$

$$e_{n+1} = e_n + h\frac{\partial f}{\partial Y}(x_n, \zeta_n)e_n + \frac{h^2}{2}Y''(\xi_n) \tag{15}$$

$$\zeta_n \text{ between } y_n(x_n) \text{ and} Y(x_n)$$

Since y_n converges to $Y(x_n)$ on $[x_0, b] \Rightarrow \dfrac{\partial f(x_n, \zeta_n)}{\partial Y} \rightarrow \dfrac{\partial f(x, Y(x))}{\partial y}$

and thus bounded in magnitude over $[x_0, b]$. Pick $h_0 > 0$ so that

$$1 + h\frac{\partial f(x_n, \zeta_n)}{\partial Y} \geq -1 \qquad x_0 \leq x_n \leq b \qquad \forall h \leq h_0$$

by assumption $\dfrac{\partial f}{\partial Y} < 0 \, \forall h$. Apply these 2 facts to (15) to get

$$|e_{n+1}| \leq |e_n| + \frac{h^2}{2}|Y''(\xi_n)|$$

and then by induction show $|e_n| \leq |e_0| + \dfrac{h^2}{2}[|Y''(\xi_0)| + \cdots + |Y''(\xi_{n-1})|]$

which leads to result:

$$\boxed{|Y(x_n) - y_n(x_n)| \leq |e_0| + \frac{h}{2}(x_n - x_0) \max_{x_0 \leq x \leq x_n} |Y''(x)|}$$

□

Next we consider the "stability of solutions". An IVP may posess stable or unstable solutions, or both. Whether it does depends on $f(x, Y)$, and on the initial data. If the IVP problem is approximated using a numerical scheme, we would like the numerical scheme to have the approximate solution behave as the IVP solutions being studied. However, it is possible that the discretization, may behave in ways that are different from the IVP being approximated...the discretization depends not only on f and on the initial data, but also on how the equation is discretized, on how big the step size is, on how the dependent and independent variables are represented mathematically and on the machine. The study of convergence enabled us to determine whether the approximate solution approached the real solution of the IVP as h got smaller for general choices of initial data and at what rate (see0.1.5). Stability will detemine whether systematic errors (which are usually very small) such as round-off and/or uncertainty in the initial data will make arbitrarily close solution paths separate from each other at a rate greater than linear. For the IVP, this is considered its innate behavior and it is important to know whether what you're approximating has this behavior. For the numerical approximation, we want to know if what we are seeing is due to the innate behavior of the IVP or due to using an inappropriate scheme for the numerical approximation of the IVP.

Remark Numerical instability usually leads to spectacularly bad results, i.e. code crashes. But if we had to rank what's worse, lack of convergence or instability, lack of convergence is actually worse: the reason is that lacking convergence means that we are not solving the IVP we think we're solving but some other IVP! Computationally, also, when instabilities manifest themselves they usually force you to do something about it. But sometimes non-convergent numerical schemes are happy to provide you with all sorts of answers and you'd not suspect anything wrong...well, till you kill someone by solving the wrong equation in the first place.

Stability of IVP

$$\text{Take } \begin{cases} y' = f(x, y) + \delta(x) \\ y(x_0) = Y_0 + \varepsilon \end{cases} \tag{16}$$

$$\text{(17)} \qquad \begin{cases} Y' = f(x, Y) \\ Y(x_0) = Y_0 \end{cases}$$

$f(x, y)$ continuous and satisfies a Lipschitz Condition (L.C.). Also, assume $\delta(x)$ continuous for all (x, y) interior points to convex set. We show that solution to (16) is controlled by ε and is unique. Then, we'll use this to infer stability and well-poseness. Note: ε is a perturbation parameter.

Theorem. (Perturbed Problem): Assume hypothesis on $f(x, y)$ as above and hypothesis on $\delta(x)$ as above. (16) has solution $Y(x; \delta, \varepsilon)$ on $[x_0 - \alpha, x_0 + \alpha], \alpha > 0$, uniformly $\forall \varepsilon$ perturbations and $\delta(x)$ that satisfy

$$|\varepsilon| \leq \varepsilon_0 \qquad ||\delta||_\infty \leq \varepsilon_0$$

for ε_0 sufficiently small. In addition, if $Y(x)$ is solution of (17), then

$$\max_{|x - x_0| \leq \alpha} |Y(x) - Y(x; \delta, \varepsilon)| \leq k[|\varepsilon| + \alpha||\delta||_\infty]$$

$$k = \frac{1}{1 - \alpha L}$$

Proof: exercise. This theorem assumes that αL are sufficiently small. You'll also need

$$1 + r + r^2 + \ldots = \frac{1}{1 - r}$$

for $r < 1$ □

Studying the stability of an equation enables us to tell whether

1) Forward Euler (or any other numerical scheme) produces approximate solutions that are close enough to $Y(x)$, the exact solution. Very oftenly we need to determine how close as well.

2) To identify whether equation is "STIFF" and/or "ILL-CONDITIONED" (this is a topic considered a little later, but for now just think of "Stiff" as "very difficult" to solve numerically.

For simplicity, consider ε perturbations (the $\delta(x) \pm 0$ case is a little more involved but enters as per previous theorem).

Take $Y_0 \to Y_0 + \varepsilon$:

$$\begin{aligned} Y'(x;\varepsilon) &= f(x, Y(x;\varepsilon)) \qquad x_0 - \alpha \le x \le x_0 + \alpha \\ Y(x_0;\varepsilon) &= Y_0 + \varepsilon. \end{aligned} \tag{18}$$

Subtract (18) from (17): let $Z(x) = Y(x,\varepsilon) - Y(x) \Rightarrow Z(x_0,\varepsilon) = \varepsilon$

then

$$Z'(x;\varepsilon) = f(x, Y(x;\varepsilon)) - f(x, Y(x)) \approx \frac{\partial f}{\partial Y}(x, Y(x)) Z(x;\varepsilon) \tag{19}$$

Is $Y(x;\varepsilon)$ close to $Y(x)$ as $x \to \infty$? Maybe, if ε is small enough and $[x_0 - \alpha, x_0 + \alpha]$ small.

We can solve (19)

$$Z(x;\varepsilon) \approx \varepsilon e^{\int_{x_0}^{x} \frac{df}{dY}(t, Y(t))dt}$$

If $\dfrac{\partial f}{\partial Y}(t, Y(t)) \le 0 \quad |x_0 - t| \le \alpha \Rightarrow z(x,\varepsilon)$ remains bounded by ε as x increases.

$$\Rightarrow \text{WE SAY THAT (2) is WELL-CONDITIONED !}$$

Example:

$$\begin{cases} Y' = \lambda Y + g(x) \quad \lambda > 0 \quad \lambda \text{ constant.} \\ Y(0) = Y_0 \end{cases}$$

$\dfrac{\partial f}{\partial Y} = \lambda$ and $z(x,\varepsilon) = \varepsilon e^{\lambda x}$ exactly

$\therefore$ perturbations grow large as x increases.

$\Rightarrow$ "ILL-CONDITIONED" and "STIFF" if λ LARGE

Example:

$$\begin{cases} Y' = 100Y - 101e^{-x} \\ Y(0) = 1 \end{cases} \tag{20}$$

$$\text{Solution } Y(x) = e^{-x}$$

take perturbations

$$Y' = 100Y - 101e^{-x} Y(0) = 1 + \varepsilon$$

$$\text{Solution: } Y(x;\varepsilon) = e^{-x} + \varepsilon e^{100x}$$

which rapidly departs from true solution $Y(x)$. We say that (20) is ill-conditioned.

For well-conditioned we require that $\int_{x_0}^{x} \frac{\partial f}{\partial Y}(t, y(t))dt$ be bounded from above by 0 or a small positive number as x increases $\Rightarrow Z(x;\varepsilon)$ will be bounded by constant times ε.

If $\frac{\partial f}{\partial Y} \leq 0$ but large $\Rightarrow$ call ODE STIFF

and these cases present problems, numerically.

□

Stiffness is a qualitative assessment of an ODE or a system of them. In a single ODE stiffness can be assessed as we did above by having some good bounding criteria for f, and it is the bounding value that determines how "stiff" the ODE is. In a system of ODE's stiffness not only brings into play the size of each f but also the relative size of each of these. That is, in addition to the value of each f_i, what also comes into play is the wide discrepancy in the rate of change of the f_i's. If you think of x as a time parameter and can bound the rate of change of each f_i by a constant, say, then if these constants are very disparate we say the ODE system is stiff and it manifests its complication in the existence of a wide span of time scales in the behavior of the solution.

Stability Analysis of Forward Euler Scheme

first we motivate problem with important example:

Example) $\begin{cases} y' = \alpha y, & \alpha \quad \text{constant.} \\ y(0) = y_0 > 0 \end{cases}$

It has the exact solution: $y(x) = e^{\alpha x} y_0$. Assume $x > 0$.

Using Forward Euler: $y_{n+1} = (1+\alpha h)y_n \rightarrow y_{n+1} = (1+\alpha h)^n y_0$ take h constant for simplicity.

Case (a):	$-1 < \alpha h < 0$	$\Rightarrow$ solution positive and approaching 0.
Case (b):	$\alpha h < -1$	$\Rightarrow$ sign of solution alternatives as n increases.
Case (c):	$\alpha h > 0 \Rightarrow$	increases at each step

□

Definition: In general, for $n = 0, 1, 2, \cdots$ a scheme of the form $y_{n+1} = g(y_n, y_{n-1}, \cdots y_0)$ is said to be "explicit." If $y_{n+1} = g(y_{n+1}, y_n, y_n \cdots y_0)$ then scheme is said to be "implicit." Example) Forward Euler is said to be an "explicit" scheme because each y_{n+1} can be solved in terms of y_n.

Stability Analysis for Forward Euler:

Consider

$$\text{(21)} \qquad \begin{cases} z_{n+1} = z_n + h[f(x_n, z_n) + \delta(x_n)] & 0 \le n \le N(h) - 1 \\ z(0) = y_0 + \varepsilon \end{cases}$$

and

$$\text{(22)} \qquad \begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = y(x_0) \end{cases}$$

Look at $\{z_n\}$ and $\{y_n\}$ as $h \rightarrow 0$ and as n increases:

let

$$e_n = z_n - y_n \quad , n \ge 0 \Rightarrow \quad e_0 = \varepsilon$$

Subtract (22) from (21):

$$e_{n+1} = e_n + h[f(x_n, z_n) - f(x_n, y_n)] + h\delta(x_n)$$

has same form as equation (15) $\therefore$

$$\max_{0 \le n \le N(h)} |z_n - y_n| \le e^{(b-x_0)L}|\varepsilon| + \Big[\frac{e^{(b-x_0)L} - 1}{L}\Big]||\delta||_\infty$$

$\therefore \exists\, k, k_2$ independent of h with

$$\max_{0\le n\le N(h)} |z_n - y_n| \le k_1|\varepsilon| + k_2||\delta||_\infty.$$

Effect of Rounding Errors:

take ρ_n to be the local rounding error and let

$$\tilde{y}_{n+1} = \tilde{y}_n + hf(x_n, \tilde{y}_n) + \rho_n \quad n = 0, 1 \ldots N(h) - 1 \tag{23}$$

$$\text{let} \quad \begin{cases} \tilde{y}_n \text{ are finite precision numbers} \\ y_n \text{ be exact arithmetic.} \\ \rho(h) \equiv \max\limits_{0\le n\le N(h)-1} |\rho_n| \end{cases}$$

$$Y_{n+1} = Y_n + hf(x_n, y_n) + \frac{h^2}{2}y''(\xi_n) \tag{24}$$

Subtract (24) from (23): $\tilde{e}_{n+1} = \tilde{e}_n + h[f(x_n, Y_n) - f(x_n, \tilde{y}_n)] + h\tau_n - \rho_n$

where $\tilde{e}_n = Y(x_n) - \tilde{y}(x_n) \quad and \quad \tau_n \equiv \dfrac{h}{2}Y''(\xi_n)$

Use same arguments as before, but let $\tau_n - \rho_n/h$ replace τ_n of before

$$|\tilde{e}_n| \le e^{(b-x_0)L}\left|Y_0 - \tilde{y}_0\right| + \left|\frac{e^{(b-x_0)L} - 1}{L}\right|\left[\tau(h) + \frac{\rho(h)}{h}\right].$$

On a finite precision machine $\rho(h)$ will not decrease as $h \to 0 \ldots$ it'll remain finite and approximately constant. Take

$$u \equiv \frac{\rho(h)}{||Y||_\infty} \text{ then}$$

$$|\tilde{e}_n| \le c\left[\frac{h}{2}\big|\big|Y''\big|\big|_\infty + \frac{u}{h}\big|\big|Y\big|\big|_\infty\right] \equiv E(h)$$

We can find the h which minimizes the error. Call it $h^\star$. To find, set $\dfrac{dE(h^\star)}{dh} = 0$ and h^* corresponding to minima.

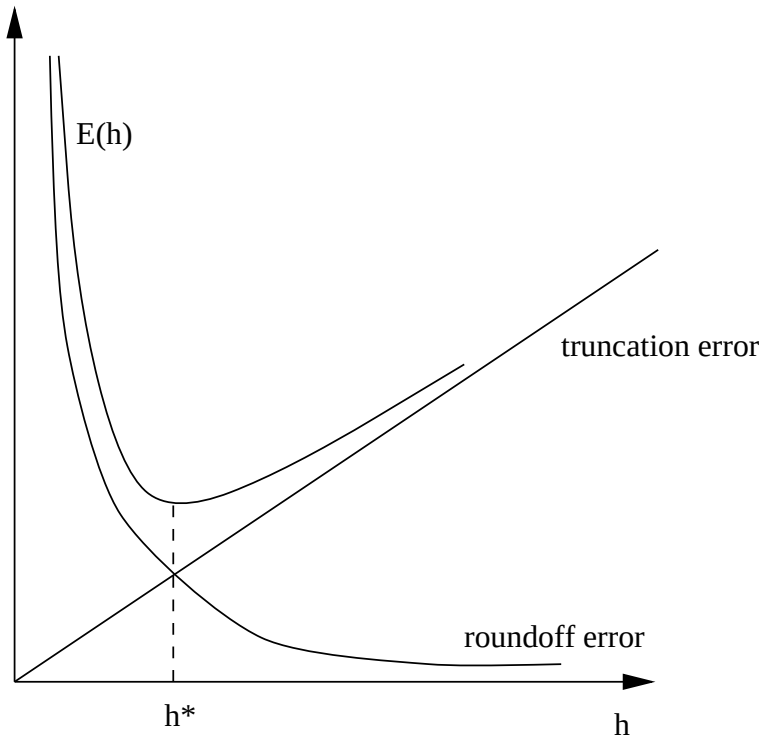

Figure 5: Effect of Rounding Errors on Forward Euler

See Figure 5.

Asymptotic Error Analysis

Recall that if $B(x,h)$ is a function defined for $x_0 \le x \le b$, for sufficiently small $h \Rightarrow$

$B(x,h) = \mathcal{O}(h^p)$ $\quad p > 0$ near $\exists$ a constant c such that
$|B(x,h)| \le ch^p$ $\quad x_0 \le x \le b$

Theorem. (Euler Error): Assume $Y(x)$ = solution of ODE and 3 times continuously differentiable. Assume f_y and f_{yy} are continuous and bounded for $x_0 \le x \le b, -\infty < y < \infty$. Let the initial value $y_n(x_0)$ satisfy

$$Y_0 - y_h(x_0) = \delta_0 h + \mathcal{O}(h^2)$$

usually this error is machine precision or zero.

Then the error in Forward Euler's $y_{n+1} = y_n + hf(x_n, y_n)$ satisfies

$$Y(x_n) - y_h(x_n) = D(x_n)h + \mathcal{O}(h^2)$$

where $\begin{cases} D'(x) = f_y(x, Y(x))D(x) + \frac{1}{2}Y''(x) \\ D(x_0) = \delta_0 \end{cases}$

Proof. before the proof, let's do an example:

Example: $y' = -y \quad y(0) = 1$

has solution $Y(x) = e^{-x}$. The $D(x)$ equation is

$$\begin{cases} D' = -D + \frac{1}{2}e^{-x} \\ D(0) = 0 \end{cases}$$

$$\therefore D(x) = \frac{1}{2}xe^{-x}$$

So the error for Forward Euler is $Y(x_n) - y_h(x_n) \approx \frac{h}{2}x_n e^{-x_n}$

$$\therefore \text{the relative error } \frac{Y(x_n) - y_h(x_n)}{Y(x_n)} \approx \frac{h}{2}x_n$$

Calculation shows that the error is linearly proportional to h.

Remark. We say Euler is an "$\mathcal{O}(h)$ method"

Proof. Use Taylor's

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \frac{h^2}{2}Y''(x_n) + \frac{h^3}{6}Y'''(\xi_n) \tag{25}$$
$$\text{for some } x_n \leq \xi_n \leq x_{n+1}$$

We have

$$Y'(x) = f(x, Y(x)) \tag{26}$$

and

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{27}$$

Subtract (27) from (25) and use (26)

$$e_{n+1} = e_n + h\big[f(x_n, Y_n) - f(x_n, y_n)\big] + \frac{h^2}{2}Y''(x_n) + \frac{h^3}{6}Y(''')(\xi_n). \tag{28}$$

Continuity in $f(x, y)$ allows us to expand around Y_n:

$$f(x_n, y_n) = f(x_n, Y_n) + (y_n - Y_n)f_y(x_n, Y_n) + \frac{1}{2}(y_n - Y_n)^2 f_{yy}(x_n, \xi)$$
$$\text{for } \xi_n \text{ between } y_n \text{and } Y_n.$$

Plug into (28)

$$e_{n+1} = [1 + hf_y(x_n, Y_n)]e_n + \frac{h^2}{2}Y''(x_n) + B_n$$
$$B_n = \frac{h^3}{6}Y^{(''')}(\xi_n) - \frac{1}{2}hf_{yy}(x_n, \xi_n)e_n^2 = 0(h^3, he_n^2) \tag{29}$$

Neglecting B_n, then the error

$$\begin{cases} e_{n+1} \approx \underbrace{[1 + hf_y(x_n, Y_n)]e_n + \frac{h^2}{2}Y''(x_n)}_{gn} & \text{So } e_n = \mathcal{O}(h) \\ \text{with} \\ e_0 = \delta_0 h (\text{since } Y_0 - y_h(x_0) = \delta_0 h + \mathcal{O}(h^2)) \end{cases}$$

and $e_{n+1} = D(x_n)h + \mathcal{O}(h^2)$

Now, need to show that g_n is principal part of error e_n. Let

$$k_n = e_n - g_n. \quad k_0 = e_0 - g_0 = \mathcal{O}(h^2) \text{ by } \begin{cases} Y_0 - y(x_0) = \delta_0 h + \mathcal{O}(h^2) \\ g_0 = \delta_o h \end{cases}$$

$$\begin{aligned} k_{n+1} &= [1 + hfy(x_n, Y_n)]k_n + B_n \\ k_{n+1} &\leq (1 + hL)|k_n| + O(h^3) \\ \text{but } |k_n| &= \mathcal{O}(h)^2 \text{at the very least } \therefore \\ e_n = g_n + k_n &= [h\mathcal{O}(x_n) + \mathcal{O}(h^2)] + O(h^2) \end{aligned}$$

□

Forward Euler is a simple but not always appropriate scheme for solving ODE'S. Let's consider some alternatives:

0.1.6 Taylor-series Method

Consider the "Taylor Series" interpretation of problem (see 0.1.3). In this method, we assume (or determine, which is the right thing to do) that all the necessary partial derivatives exist.

Forward Euler was a first order method and in this light, results from keeping the first term of Taylor series. Why not keep the p^{th} order term in Taylor series? Is this "better" than Forward Euler? In what ways? Clearly, it is more expensive computationally, so if we are going to develop a technique that is computationally more expensive we'd better make sure we find out in what circumstances it works, if at all.

First, look at how this works by example:

Example: Solve $\begin{cases} x' = \cos t - \sin x + t^2 \\ x(-1) = 3 \end{cases}$

recall $x(t+h) = x(t) + hx'(t) + \frac{1}{2}h^2x''(t) + \frac{1}{6}h^3x'''(t) + \frac{1}{24}h^4x''''(t) + \cdots$

$$\begin{aligned} x'' &= -\sin t - x'\cos x + 2t \\ x''' &= -\cos t - x''\cos x + (x')^2 \sin x + 2 \\ x'''' &= \sin t + (x')^3\cos x + 3x'x''\sin x - x'''\cos x \end{aligned}$$

let's stop there. So we say that we're constructing "an approximation of order 4," which means that THE LOCAL TRUNCATION ERROR is $\mathcal{O}(h^5)$: as $h \to 0$ the local error is proportional to Ch^5 (we don't know what is C and how big it is).

Algorithm

input M (steps), t_0 and t_f

output (x_n, t_n); compute $\begin{cases} h = (t_f - t_0)/M, \\ t_k = t_0 + kh \qquad k = 0 \cdots M+1 \end{cases}$

for $k = 1 : M$

$$\begin{aligned} x' &= \cdots \\ x'' &= \cdots \\ x''' &= \cdots \\ x'''' &= \cdots \\ x_{k+1} &= x_k + h\Big(x' + \frac{h}{2}\Big(x'' + \frac{h}{3}\Big(x''' + \frac{h}{4}x''''\Big)\Big)\Big) \end{aligned}$$

□

In above example we can calculate local truncation error:

$$E_n \approx \frac{1}{(n+1)!}h^{n+1}x^{(n+1)}(t+\theta h) \quad 0 < \theta < 1$$

in above example $n = 4$. Could use simple finite differences:

$$E_4 \approx \frac{1}{5!}h^5\left[\frac{x^{(''''')}(t+h) - x^{(''''')}(t)}{h}\right] = \frac{h^4}{120}\Big[x^{(''''')}(t+h) - x^{(''''')}(t)\Big]$$

Pros and Cons

1) Number of operations can be large. This is not always a problem these days considering how cheap and fast computers are at present. In general the step size can be made larger the higher the order of the Taylor scheme. But a computational count will tell you whether it is more effective to compute a lot at each step, and take larger steps, or compute little at each step and make due with smaller step sizes. Usually it is more advantageous to go with good lower order method and small steps, but this depends on the problem..

2) Need to know about smoothness of solution. This is neither a pro or a con since you should always know this, but it is senseless to use a high order method when the solution has unbounded high order derivatives.

3) Can use symbolic program to compute all the series expansion stuff reducing programming errors, so in principle, it is not too hard to compute high order derivatives required.

4) For oscillatory functions, a high order Taylor seems a good choice, but can lead to problems of loss of significance in the computation if not programmed carefully.

In general the high order Taylor Method is used "in special situations" such as when we want the quantity at a point with low truncation error. □

QUESTION: Do we always seek to use a high-order method? What does the **ORDER** of a method have to do with **accuracy**?

First off, the "order" is a property of a particular scheme. The "accuracy" merely refers to how "close" a particular approximation is to the exact answer. It is actually up to you to define precisely to your audience what you mean by "accurate": it may simply mean how far the approximation is to an exact answer, in some appropriate norm. But it can be more complicated: as we will later see in the context of solving wave equations, accuracy may be favoring the overall shape of the wave, rather than its magnitude, or its overall phase, etc).

Suppose we define by accuracy merely the Euclidean distance between the approximate and exact solution at a single point. Then, provided that a scheme converges, the order of a method will tell you the rate at which, in

the limit as the step-size goes to zero, the approximation approaches the exact answer at any or all points in the domain of the problem. A high order method will approach the exact answer at a faster rate, if you make your step-size smaller. Now, suppose you have two different convergent schemes with different rates of convergence. Is the one with higher order of convergence more "accurate"?? The answer is "it depends".

YES, if the step size is the same in both methods, the higher-order method will be more accurate (assuming there are no round-off errors).

NO, not necessarily. If you compute with the low order method using a small h and compute with the high order method with a large h it is possible to get higher accuracy with the low order method.

Usually, high order methods are more involved computationally and low order methods less so. Ultimately, you desire high accuracy and you're willing to pay for it in terms of time in computing. If a low order method takes little computer time per time step, compared to a higher order method, the amount of computer time required to get an answer depends on the per-step size time, times the number of steps required. Hence, for a given accuracy you can (and should) always estimate which method is best to use. Now, in these days of fast computers, sometimes a low order method is preferred. But back in the old days, when computers were really room-fulls of people operating adding machines and/or slide rulers in a systematic way (cf. Virtual Museum of Computing), high order methods were often sought. As you might imagine, the "step-size" in problems solved in this fashion was required to be quite large.

Another reason for which a low order method and/or an explicit method is preferred to an implicit method has to do with the other issue in computational implementation: storage use. A high order method and/or some implicit methods will require vast amounts of memory usage...in some cases, more than you have. Often times the trade off amounts to giving up speed in the computation in the interest of memory, which might force you to use a lower order method and/or an explicit method.

The storage vs. speed on machines has become a more important issue these days, of parallel computing. A scheme that requires a great deal of communication between processors will tax most significantly the gains in

speed-up possible in parallel processing. A low order method usually will be more local, leading then to less communication between processors.

In summary, "Order" is not the End of All Things. Decide on what accuracy you're willing to live with and what your criteria is going to be. Then, compare schemes for the approximation and include in your decision of which one to pick the one that will deliver the most robust results with a level of computational efficiency that you are willing to live with: this estimate should be done WAY AT THE BEGINNING. There is no need to gamble gobbs of time on an effort that you could have otherwise determined (with little effort) is less than satisfactory.

0.1.7 Trapezoidal Rule

Adopt the "integral" interpretation of problem (see 0.1.3). Recall Forward Euler approximates derivative by a constant at x_n

$$Y_{n+1} = Y_n + \int_{x_n}^{x_{n+1}} f(t, Y(t))dt \approx Y_n + hf(x_n, Y(x_n))$$

The Trapezoidal rule estimates "height" of box by average of f at x_n and x_{n+1} :

$$\boxed{y_{n+1} = y_n + \tfrac{1}{2}h\big[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\big]} \tag{30}$$

i.e.

$$y(x) = y(x_n) + \int_{x_n}^{x} f(t, y(t))dt$$
$$\approx y(x_n) + \frac{1}{2}(x - x_n)(f(x_n, y_n) + f(x_{n+1}, y_{n+1})).$$

To find the order of method, take an exact solution

$$Y_{n+1} = Y_n + hY'(x_n) + \frac{1}{2}hY''(x_n) + \mathcal{O}(h^3)$$

and subtract (30):

$$\begin{aligned} Y_{n+1} &- \left\{ y_n + \frac{1}{2}h[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \right\} \\ &= Y_n + hY'(x_n) + \frac{1}{2}h^2 Y''(x_n) + \mathcal{O}(h^3) \\ &- \left\{ y_n + \frac{1}{2}h\Big[y'_n + [y'(x_n) + hy''(x_n) + O(h^2)] \Big] \right\} = O(h^3) \end{aligned}$$

Hence, trapezoidal is Order-2 Method. Before inferring that the error decays globally as $\mathcal{O}(h^2)$, we need to prove the method is convergent:

Theorem.

The Trapezoidal Rule is convergent.

Proof. Exercise (use strategy of multistep method considered later). □
Consider trapezoidal on model

$$\text{(31)} \qquad \begin{cases} Y' = \lambda Y \\ Y(0) = 1 \quad \text{solution } Y(x) = e^{\lambda x} \end{cases}$$

or more generally, on

$$\begin{cases} Y' = \lambda Y + g(x) \\ Y(0) = Y_0 \end{cases}$$

where $x > 0$ and λ complex.

$$\text{(32)} \qquad \therefore \begin{cases} y_{n+1} = y_n + \frac{h}{2}[\lambda y_n + g(x_n) + \lambda y_{n+1} + g(x_{n+1})] \quad n \geq 0 \\ y_0 = Y_0 \end{cases}$$

and perturbed case

$$\begin{cases} z_{n+1} = z_n + \frac{h}{2}[\lambda z_n + g(x_n) + \lambda z_{n+1} + g(x_{n+1})] \quad n \geq 0 \\ z_0 = Y_0 + \varepsilon \end{cases}$$

let $w_n = z_n - y_n$. Subtracting:

$$\begin{cases} w_{n+1} = w_n + \frac{h}{2}[\lambda W_n + \lambda W_{n+1}] \quad n \geq 0 \\ w_0 = \varepsilon \end{cases}$$

i.e. Trapezoidal rule again! Solution is what's obtained by trapezoidal on (31) except $Y_0 = \varepsilon$.

$\therefore$ Can look at (31) to assess stability:

Apply trapezoidal on (31):

$$\begin{cases} y_{n+1} = y_n + \frac{h\lambda}{2}[y_n + y_{n+1}] & n \geq 0 \\ y_0 = 1 \end{cases}$$

Consider

$$\begin{cases} Re(\lambda) < 0, \\ \lambda \text{ complex, with } \mathbb{R}e(\lambda) < 0 \Rightarrow & Y' = \lambda Y + g(x) \quad \text{well-conditioned} \\ & \text{i.e. } \dfrac{\partial f}{\partial Y} \leq 0. \end{cases}$$

In this case we expect the limiting value of the approximation to be the same as that of the solution, i.e. $\lim_{x\to\infty} Y(x) = 0$. So

$$y_{n+1} = \Big[\frac{1 + (h\lambda/2)}{1 - (h\lambda/2)}\Big] y_n \qquad n \geq 0$$

thus, by induction,

$$y_n = \Big[\frac{1 + (h\lambda/2)}{1 - (h\lambda/2)}\Big]^n y_0 \qquad n \geq 0$$

since $y_0 = 1$. with $h\lambda \neq 2$. What we want to check is to see if there are any limits imposed on h for the scheme to deliver an approximation that has the same asymptotic quality as the exact solution.

For $\mathbb{R}e(\lambda) < 0$

$$r = \frac{1 + (h\lambda/2)}{1 - (h\lambda/2)} = 1 + \frac{h\lambda}{1 - (h\lambda/2)} = -1 + \frac{2}{1 - (h\lambda/2)}$$

$$\therefore -1 < r < 1 \quad \forall h > 0 \quad \therefore \quad \lim_{n\to\infty} y_n = 0 \tag{33}$$

$\therefore$ no limitations on h in order to have boundedness of $\{y_n\}$ $\therefore$ stability of method on model equation (30) assured for $\forall h > 0$ and all (λ)
with $Re(\lambda) < 0$.

Remark. This is stronger then in most methods were stability is assured for sufficiently small h. (33) property $Ah > 0$ and $\mathbb{R}e(\lambda) < 0$ is called "A-Stability $\cdots$ important in stiff problems. (More later.)

Remark. Two asumptions lead to Trapezoidal: (A) approximate derivative by constant (B) average (not discriminate) endpoints.

$\therefore$ another possibility:

$$y'(x) \approx f(x_n + \frac{1}{2}h, \frac{1}{2}(y_n + y_{n+1})) \quad x \in [x_n, x_{n+1}]$$

leads to "implicit midpoint" method:

$$y_{n+1} = y_n + hf\left(t + \frac{1}{2}h, \frac{1}{2}(y_n + y_{n+1})\right)$$

Exercise: show that this scheme is 2^{nd} order and convergent.

0.1.8 Theta Method

This method is also known as the weighted method. Both Euler and Trapezoidal rules fit an equation of the form

$$y_{n+1} = y_n + h\big[\theta f(t_{n+1}y_{n+1}) + (1-\theta)f(t_n, y_n)\big] \quad n = 0, 1\cdots$$

where $\theta \in [0, 1]$. When $\theta = 0$(explicit), where $\theta \neq 0$ (implicit). Note that $\theta = 0$ is Euler, $\theta = \frac{1}{2}$ is Trapezoidal.

Order of Method: (Exercise) Show that the difference between the exact solution and the above approximation at $t = t_n$ is

$$(\theta - \frac{1}{2})h^2y''(t_n) + \left(\frac{1}{2}\theta - \frac{1}{3}\right)h^3y'''(t_n) + \theta(h^4)$$

hence method is order 2 for $\theta = \frac{1}{2}$ (corresponding to Trapezoidal) and otherwise is of order 1. $\square$

If we go through the usual argument (exercise), for $h > 0$ and sufficiently small, then

$$\begin{aligned} e_{n+1} = e_n + \theta h[f(t_n, y(t_n) + e_n) - f(t_n, y(t)n))] \\ +(1-\theta)h\Big[f(t_{n+1}, y(t_{n+1}) + e_{n+1}) \\ -f(t_{n+1}, y(t_{n+1}))\Big] \end{aligned}$$

$$\begin{cases} -\frac{1}{12}h^3 y'''(t_n) + \mathcal{O}(h^4) & \theta = \frac{1}{2} \\ +(\theta - \frac{1}{2})h^2 y''(t_n) + \mathcal{O}(h^3) & \theta \neq \frac{1}{2} \end{cases}$$

Now, take e_{n+1} as an unknown and apply implicit function theorem.

Ok, since f is analytic and for $h > 0$ sufficiently small, the matrix

$$I - (1-\theta)h\frac{\partial f(t_{n+1}, y(t_{n+1}))}{\partial y} \text{ is nonsingular.}$$

Then, using the implicit function theorem one can can show (try it!)

$$e_{n+1} = e_n \begin{cases} -\dfrac{1}{12}h^3 y'''(t_n) + \mathcal{O}(h^4) & \theta = \frac{1}{2} \\ +(\theta - \frac{1}{2})h^2 y''(t_n) + \theta(h^3) & \theta \neq \frac{1}{2} \end{cases}$$

Why bother with the Theta Method i.e. with θ taking any value in $[0, 1]$, not just $1/2$ and 1?

1) The concept of order is based on assumption that error is concentrated on the leading order of Taylor series expansion (on real computers, h is small, but finite). e.g. $\theta = \dfrac{2}{3}$ gets rid of $\mathcal{O}(h^3)$ while retaining $\mathcal{O}(h^2)$. Hence, for different types of $f(t, y)$ one can tune θ to control whether $\mathcal{O}(h^3)$ and higher order terms or $\mathcal{O}(h^2)$ and higher order terms contribute to the overall error when h is finite. It may be possible to choose a θ that generates a more optimal or smaller error ....

2) Theta Method is an example of a general approach to designing algorithms in which geometric intuition is replaced by Taylor series expansion. Invariably the implicit function theorem is also used in the design and analysis of scheme.

3) The $\theta = 1$ Case is very practical:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) \quad n = 0, 1, \cdots$$

This is the "Backward Euler" or "Implicit Euler" scheme, a simple yet robust method for solving STIFF ODES (Stiffness will be discussed later in detail).

4) Comparison of the Trapezoidal and Euler methods (see reftreu) will be done later, but the Euler method is more dissipative than the trapezoidal and in some problems a little more or a little less dissipation is appropriate or wanted.

0.1.9 The Runge-Kutta Family (RK)

Now we revert to a "Taylor series" interpretation of problem (see 0.1.3). We consider in some detail the EXPLICIT CASE and will make only cursory comments on the IMPLICIT Runge-Kutta methods. RK are single-step methods and can be either explicit or implicit. Later we'll consider multi-step methods.

Perhaps the most popular ODE integrator around, because it is explicit, easily programmable, of high order, and applicable to even mildly stiff problems: 4^{th}-order ERK (Explicit Runge Kutta) or ERK4. The 2^{nd} order $\rightarrow ERK2$ is known as Heun's Method and is also popular but used less often.

Pro's:	simple to program
	truncation error can be straight forward to control
	Good packages available (see Netlib, NIST, even matlab).
	Decent stability regimes.
Con's:	Requires many more evaluations of derivative to obtain same accuracy as compared to multistep methods.
	Only appropriate for non-stiff and very mildly stiff equations.
	Mild dissipation (invariably one uses small-order ERK method).

All ERK's are written as

$$y_{n+1} = y_n + h \sum_{j=1}^{\nu} b_j f\left(x_n + c_j h, y(x_n + c_j h)\right) \equiv y_n + hF(x_n, y_n, h, f)) \, n \geq 0,$$

where $c_j's$ are between 0 and 1. The whole point is to specify y at the locations $x_n + c_1h, x_n + c_2h, \cdots, x_n + c_\nu h$ and find the corresponding b_j. These $b_j's$ must sum to 1 so that we get a weighted average. What's the criteria? The choice of the entries in the vectors **b** and **c** make the leading terms in the truncation error equal to zero. Additionally, we want

$$F(x, Y(x), h; f) \approx Y'(x) = f(x, Y(x)), \text{for small } h$$

Example: Take trapezoidal with an Euler predictor step

$$y_{n+1} = y_n + \frac{h}{2}\Big[f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))\Big] \tag{34}$$

or $y_{n+1} = y_n + hF$ What's F? See Figure 6.

$$\therefore F \text{ is average slope on } [x, x+h]!$$

Example) Another method based on average slope is

$$y_{n+1} = y_n + hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(x_n, y_n)\right) \tag{35}$$

$$\text{Here } F = f\left(x + \frac{1}{2}h, y_n + \frac{1}{2}hf\right)$$

both (34) and (35) are 2^{nd} order.

To illustrate general procedure: ERK2 "Heun's Method"

$$Y(x+h) = Y(x) + hY'(x) + \frac{h^2}{2!}Y''(x) + \frac{h^3}{3!}Y'''(x)\cdots \tag{36}$$

and

$$\begin{cases} Y' = f \\ Y'' = f_x + f_Y y' = f_x + f_Y f \\ Y''' = f_{xx} + f_{xY}f + (f_x + f_Y f)f_Y + f(f_{xY} + f_{YY}f) \\ \text{etc.} \end{cases}$$

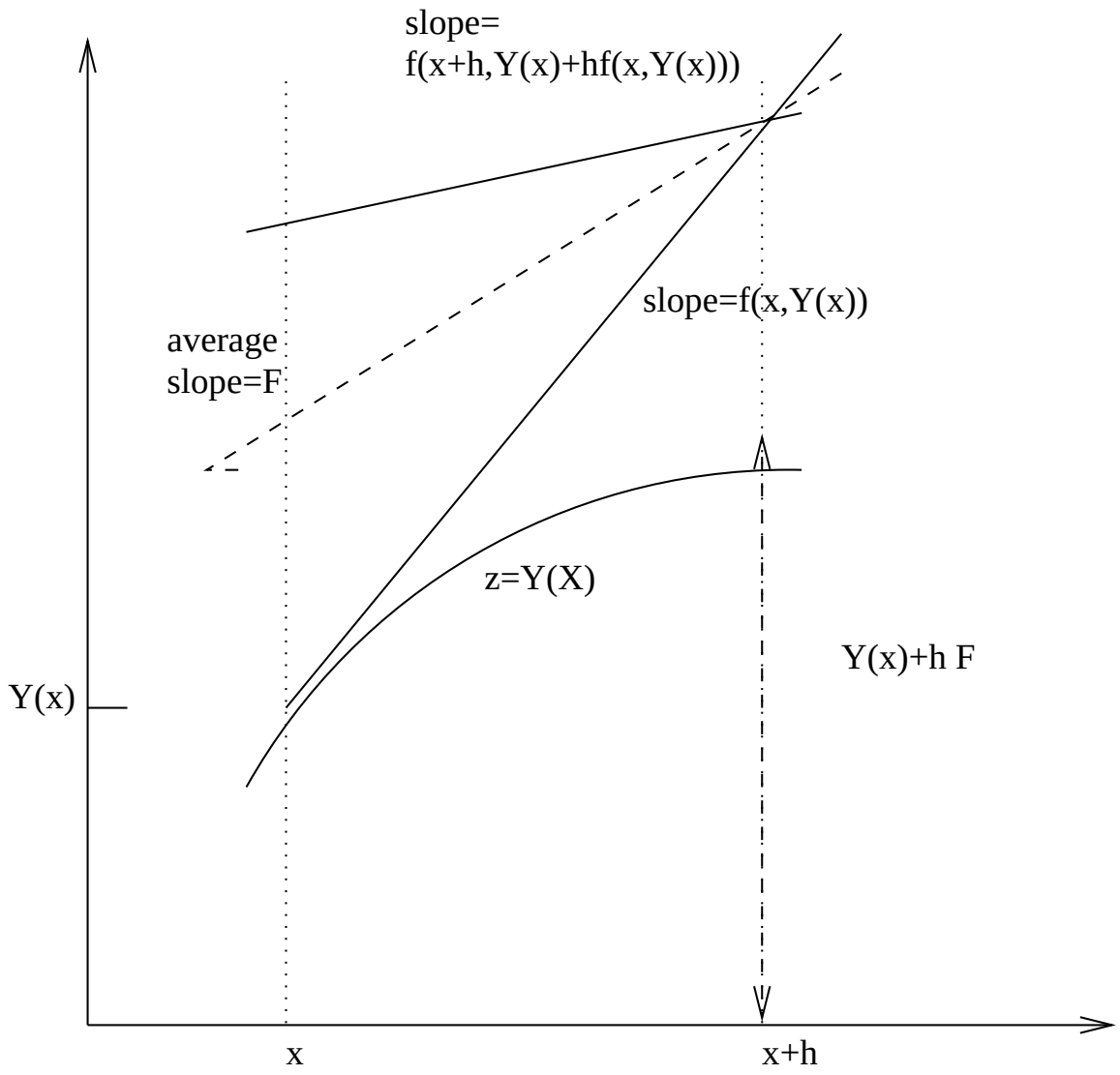

Figure 6: Geometrical interpretation of average slope

so (36) can be written as

$$Y(x+h) = Y + hf + \frac{1}{2}h^2(f_x + f_Y f) + \mathcal{O}(h^3) \tag{37}$$

$$= Y + \frac{h}{2}f + \frac{1}{2}h[f + hf_x + hff_Y] + \mathcal{O}(h^3) \tag{38}$$

but note that $f(x+h, Y+hf) = f + hf_x + hff_Y + \mathcal{O}(h^2)$

so substitute in (36)

$$Y(x+h) = Y + \frac{1}{2}hf + \frac{h}{2}f(x+h, Y+hf) + \mathcal{O}(h^3)$$

$$\left[\begin{array}{c} \therefore y(x+h) = y + \dfrac{h}{2}f + \dfrac{h}{2}f(x+h, y+hf) + \mathcal{O}(h^3) \\ F_1 = hf \quad F_2 = hf(x+h, f+F_1) \\ \text{is an ERK2 method} \\ \text{known as } \underline{\text{Heun's Method}}. \end{array}\right.$$

ERK4 Classical

$$y(x+h) = y(x) + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

$$\text{with} \begin{cases} F_1 = hf(x,y) \\ F_2 = hf(x + \frac{1}{2}h, y + \dfrac{1}{2}F_1) \\ F_3 = hf(x + \frac{1}{2}h, y + \dfrac{1}{2}F_2) \\ F_4 = hf(x+h, y+F_3) \end{cases}$$

General Method:

$$y_{n+1} = y_n + h\sum_{j=1}^{\nu} b_j f(x_n + c_j h, y(x_n + c_j h)) \quad n = 0, 1 \ldots$$

let "approximant"

$$y(x_n + c_j h) = \xi_j \quad j = 1, 2, \cdots \nu$$

let

$$c_1 = 0 \quad so \quad \xi_1 = y_n$$

The idea is to express each ξ_j with $j = 2, 3, \cdots \nu$ by updating y_n with a linear combination of

$f(x_n, \xi), f(x_n, +hc_2, \xi_2) \cdots f(x_n + hc_{j-1}, \xi_{j-1})$,i.e.

$$\begin{aligned}
\xi_1 &= y_n \\
\xi_2 &= y_n + ha_{2,1} f(x_n, \xi_1) \\
\xi_3 &= y_n + ha_{3,1} f(x_n, \xi_1) + ha_{3,2} f(x_n + c_2 h, \xi_2) \\
&\vdots \quad \vdots \quad \nu - 1 \\
\xi_\nu &= y_n + h \sum_{i=1} a_{\nu,i} f(x_i + c_i h, \xi_i) \\
\Rightarrow y_{n+1} &= y_n + h \sum_{i=1}^{\nu} b_j f(x_n + c_j h, \xi_j)
\end{aligned}$$

The matrix $A \equiv A_{j,i} \quad j, i = 1, 2 \cdots \nu \quad \equiv RK$ matrix

$$\begin{aligned}
\mathbf{b} &= [b_1 b_2 \cdots b_\nu]^T \equiv RK \text{ weights} \\
\mathbf{c} &= [c_1 c_2 \cdots c_\nu]^T \equiv RK \text{ nodes}
\end{aligned}$$

and we say that method has "ν stages"

Take simple case, with $\nu = 2$. Assume f smooth, for simplicity,

$$\begin{aligned}
&\xi_1 = y_n \\
&f(x_n + c_2 h, \xi_2) = f(x_n + c_2 h, y_n + a_{21}, hf(x_n, y_n)) \\
&= f(x_n, y_n) + h \left[c_2 \frac{\partial f}{\partial x}(x_n, y_n) + a_{2,1} \frac{\partial f}{\partial y}(x_n, y_n) f(x_n, y_n) \right] + \mathcal{O}(h^2)
\end{aligned}$$

$$(39) \quad \therefore y_{n+1} = y_n + h(b_1 + b_2) f(x_n, y_n) + h^2 b_2 \left[c_2 \frac{\partial f}{\partial x} + a_{21} \frac{\partial f}{\partial y} f \right] + \mathcal{O}(h^3)$$

but we note that $Y'' = \dfrac{\partial f}{\partial x} + \dfrac{\partial f}{\partial Y} f$ from $Y' = f(x, y)$, the IVP, and exact solution

$$(40) \quad Y_{n+1} = Y_n + hf(Y_n, x_n) + \frac{1}{2} h^2 \Big[\frac{\partial f}{\partial x}(x_n, Y_n) + \frac{\partial f}{\partial Y}(x_n, Y_n) f \Big] + \mathcal{O}(h^3)$$

$\therefore$ comparing (39) and (40) gives us

$$b_1 + b_2 = 1 \quad b_2 c_2 = \frac{1}{2} \quad a_{2,1} = c_2 \tag{41}$$

So we see that a 2-stage is not uniquely defined. Popular choices are

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array} \qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

"RK Tableaux" $\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array}$

3-Stage Examples (both are 3^{rd} order)

"Classical" $\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$ System $\begin{array}{c|ccc} 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{8} & \frac{3}{8} \end{array}$

4-Stage (fourth-order) $\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$

Compare to $Y_{n+1} = y_n + \dfrac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$

$$\text{with} \begin{cases} F_1 = hf(x_n, y_n) \\ F_2 = hf\left(x_n + \frac{1}{2}h, y_n + \dfrac{1}{2}F_1\right) \\ F_3 = hf(x_n + \frac{1}{2}h, y_n + \dfrac{1}{2}F_2) \\ F_4 = hf(x_n + h, y_n + F_3) \end{cases}$$

Relation between order and number of stages:

Max stages	1	2	3	4	5	6	7	8
Max order	1	2	3	4	4	5	6	6

Not worth it usually if number of stages is much greater than max order.

IMPLICIT RK(IRK): We won't study in detail. Complicated. Used in stiff equation solutions because they exhibit superior stability properties.

Generally: $\xi_j = y_n + h\sum_{i=1}^{\nu} a_{j,i} f(x_n + c_i h, \xi_i) \quad j = 1, 2, \cdots \nu$

$$y_{n+1} = y_n + h\sum_{j=1}^{\nu} b_j f(x_1 + c_j h, \xi_j)$$

Here $A \equiv (a_{j,i})$ is no longer lower triangular.

$$\sum_{i=1}^{\nu} a_{j,i} = c_j \quad j = 1, 2, \cdots \nu \quad \text{by convention.}$$

So for $y \in \mathbb{R}^d$ we get ν coupled algebraic equations.

For references see J.C. Butcher "The Numerical Anaylsis of ODE'S," John Wiley Publ. (He's one of the world's experts).

□

RK-FEHLBERG (RKF) This is an adaptive step size method which illustrates how error control can be incorporated into RK. The technique is not only applicable to RK, though. It is based on an idea similar to Richardson extrapolation The idea is to adapt the step size to control the error and ensure error is kept within a (reasonable) specified bound, *epsilon*. It is an example of a scheme that incorporates "error control".

Most popular $\begin{cases} 5 \text{ stage} & 4^{th} \text{ order} \\ 6 \text{ stage} & 5^{th} \text{ order} \end{cases}$

General strategy:

Take $\begin{cases} Y' = f(x, Y) \\ Y(x_0) = Y_0 \end{cases}$

For presentation purposes we will assume that the schemes under consideration are explicit, single step and that the scheme for w_{n+1} has a truncation error $\tau_{n+1}(h) = O(h^m)$ Assume the scheme is of the form $\begin{cases} w_{n+1} = w_n + hf(x_n, w_n, h) \\ w_0 = Y_0 \end{cases}$

Use another scheme with $\tilde{\tau}_{n+1}(h) = \mathcal{O}(h^{m+1})$

Take $\begin{cases} z_{n+1} = z_n + hf(x_n, z_n, h) \\ z_0 = Y_0 \end{cases}$

Assume that $w_n \approx z_n \approx Y(x_n)$, i.e. assume that the schemes w and z are convergent. The if we subtract

$$\begin{aligned} Y_{n+1} - w_{n+1} &= Y_{n+1} - w_n - hf(x_n, w_n, h) \approx Y_{n+1} - Y_n - hf(x_n, y_n, h) \\ &= h\tau_{n+1}(h) \end{aligned}$$

so $\tau_{n+1} \approx \dfrac{1}{h}\left[Y_{n+1} - w_{n+1}\right] = \dfrac{1}{h}[Y_{n+1} - z_{n+1}] + \dfrac{1}{h}[z_{n+1} - w_{n+1}]$

$\therefore \tau_{n+1} \approx \tilde{\tau}_{n+1} \quad + \frac{1}{h}[z_{n+1} - w_{n+1}]$

but $\tau_{n+1} = O(h^m)$ and $\tau_{n+1} = O(h^{m+1})$ hence the major error contribution

$$\tau_{n+1} \approx \frac{1}{h}[z_{n+1} - w_{n+1}]$$

The idea is to adjust the step size to stay within a certain error bound.

Since $\tau_{n+1}(h) = \mathcal{O}(h^m) \therefore \tau_{n+1}(h) = Kh^m$

$$\tau_{n+1}(qh) \approx K(qh)^m = q^m(Kh^m) \approx q^m\tau_{n+1}(h) = \frac{q^m}{h}(z_{n+1} - w_{n+1})$$

So choose q such that

$$\frac{q^m}{h}|z_{n+1} - w_{n+1}| \approx |\tau_{n+1}(qh)| \leq \varepsilon$$

or

$$q \leq \left(\frac{\varepsilon h}{|z_{n+1} - w_{n+1}|}\right)^{\frac{1}{m}}$$

□ Hence, q is a multiplicative factor that scales the time step h, ϵ is the error tolerance and is a specified input to the code.

You might be wondering about the fact that in the above expression z_{n+1} and w_{n+1} appear and these are quantities that are sought after. What's done

algorithmically is not unique: one possibility is to take a step and produce some proxi z_{n+1} and w_{n+1}. Make the estimate as per above equation, and if it is not satisfied, reduce q till it is. Then do the real time step.

Note, also, that on implementation, it is possible for the above condition not to be satisfied (either because you chose an ϵ that is unreasonable small, or because the method should not be applied to the IVP in the first place). Hence, an escape sequence should be supplied in the algorithm so that the user gets a warning of the method's failure.

An implementation of this is ERKF4: use a 5^{th} order RK to estimate the 4^{th} order RK (there's one for 5^{th}-order equations as well and it is easily derivable.)

ALGORITHM

INPUT x_0, b, initial condition α, TOL, h_{max}, h_{min}
OUTPUT x, y, h

Step 1 $x = x_0$; $y = \alpha, h = h_{max}$, FLAG=1; output(x, y);
Step 2 While (FLAG=1) do Step 3-11
Step 3

$$
\begin{aligned}
K_1 &= hf(x, y) \\
K_2 &= hf(x + \frac{1}{4}h, y + \frac{1}{4}K_1) \\
K_3 &= hf\left(x + \frac{3}{8}h, y + \frac{3}{32}K_1 + \frac{9}{32}K_2\right) \\
K_4 &= hf\left(x + \frac{12}{13}h, y + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right) \\
K_5 &= hf\left(x + h, y + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right) \\
K_6 &= hf\left(x + \frac{h}{2}, y - \frac{8}{27}K_1 + 2K_2 - \frac{3544}{2565}K_3\right. \\
&+ \left.\frac{1859}{4104}K_4 - \frac{11}{40}K_5\right)
\end{aligned}
$$

$$
R = \frac{1}{h}\left|\frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6\right|
$$

% Note: $R = \left|\tilde{y}_{n+1} - y_{n+1}\right| \dfrac{h}{2}$

Step 5 $\delta = 0.84\ (\mathrm{TOL}/\mathrm{R})^{\frac{1}{4}}$

Step 6 if $R \leq$ TOL do steps 7&8

Step 7 $x = x + h$
$y = y + \dfrac{25}{216}K_1 + \dfrac{1408}{2565}K_3 + \dfrac{2197}{4104}K_4 - \dfrac{1}{5}K_5$

Step 8 output (x, y, δ)

Step 9 if $\delta \leq 0.1$ then $h = 0.1h$
else if $\delta \geq 4$ then set $h = 4h$
else set $h = \delta h$ % (Calc new h).

Step 10 if $h > h_{max}$ then $h = h_{max}$

Step 11 If $x \geq b$ the FLAG $= 0$
else if $x + h > b$ then $h = b - x$
else if $h < h_{min}$ then
set FLAG $= 0$
output ('minimum x step is exceeded').

Step 12 STOP, END

□

Two final remarks are in order:

1) the formal introduction of the method in a schematic way is meant to convey that the trick is general and can be constructed using other schemes, in addition to RK. The overall choice of methods to combine should be dictated by the overall goal: to exploit adaptivity to make the computation more efficient, and to make codes that are more robust and less prone to users' bad choice of time stepping parameters.

2) All too often scientists will try to use adaptivity in step size to try to

circumvent a problematic nature of an IVP. Adaptivity is useful when the problem exhibits solutions that have periods of high activity, followed by periods of low activity. It is important that you learn to recognize the difference between an IVP that is STIFF and one that merely has periods of heightened activity. It is common for people to think that one can still use an explicit method with adaptivity to counteract the stiffness of an IVP. This is not true in general, although it might work in some circumstances. Adaptivity might force a method to stay within its stability region, by making h small enough. It works properly if the stability range gets significantly smaller and larger as the code steps through the approximate solution. It does not work, when the code is forced to use a small step and then is forced to keep such small step for the rest of the integration interval: you're better off coding something up that is simpler and more robust and forego adaptivity, since it brings no benefit. And of course, it goes without saying, that if you choose a method for which no h value could produce a stable approximation, then adaptivity is not going to help things...

Convergence Analysis for ERK

$$y_{n+1} = y_n + hF(x_n, y_n, h; f) \quad n \geq 0$$

Define the truncation error as

$$T_n(Y) = Y(x_{n+1}) - Y(x_n) - hf(x_n, Y(x_n), h; f) \quad n \geq 0$$

and

$$T_n(Y) = h\tau_n(Y)$$

For convergence we require that $\tau_n(Y) \to 0$ as $h \to 0$.

$$\tau_n = \frac{Y(x_{n+1}) - Y(x_n)}{h} - F(x_n Y(x_n), h; f)$$

then require

$$F(x, Y(x), h; f) \to Y'(x) = f(x, Y(x)) \text{ as } h \to 0$$

$$\text{Let } \delta(h) = \max_{\substack{x_0 \leq x \leq b \\ -\infty < y < \infty}} \left| f(x, y) - F(x, y, h; f) \right|$$

F is picked so that $\delta(h) \to 0 \quad$ as $h \to 0$ "CONSISTENCY CONDITION" (require Lipschitz constant to be defined)

$$|F(x,y,h;f) - F(x,z,h;f)| \le L|y-z|$$
$$\text{for all } x_0 \le x \le b \quad \& \quad -\infty < y < \infty \text{ and small } h.$$

For a particular ERK2 from before:

Example:

$$\begin{aligned}
\Big|F(x,y,h;f) &- F(x,z,h;f)\Big| = \left|f\left(x+\frac{h}{2}, y+\frac{h}{2}f(x,y)\right) - f\left(x+\frac{h}{2}, z+\frac{h}{2}f(x,z)\right)\right| \\
&\le K\left|y - z + \frac{h}{2}\left[f(x,y) - f(x,z)\right]\right| \le K\left(1+\frac{h}{2}K\right)|y-z| \\
\text{choose } L &= K\left(1+\frac{1}{2}K\right) \qquad \text{for } h \le 1
\end{aligned}$$

Theorem (Error and Convergence):

Assume Runge-Kutta $\quad y_{n+1} = y_n + hF$ satisfies Lipschitz Constant on $F \Rightarrow$ for IVP the solution $\{y_n\}$ satisfies

$$\begin{aligned}
\max_{x_0 \le x_n \le b} |Y(x_n) - y_n| &\le e^{(b-x_0)L}|Y_0 - y_0| + \Big[\frac{e^{(b-x_0)L}-1}{L}\Big]\tau(h) \\
\tau(h) = \max_{x_a < x_n \le b} &|\tau_n(Y)|
\end{aligned}$$

If the consistency condition $\delta(h) \to 0$ as $h \to 0 \Rightarrow \{y_n\} \to Y(x)$

Proof: exercise. □

Corollary If the Runge-Kutta has truncation error $\mathcal{O}(h^{m+1}) \Rightarrow$ rate of convergence of $\{y_n\}$ to $Y(x)$ is $\mathcal{O}(h^m)$

Proof: exercise. □

We will defer discussion of stability to after we talk about multistep methods.

0.1.10 Multi-step Methods

Multi-step methods come from the Quadrature Interpretation of original problem (see 0.1.3). These schemes use a number of previously computed approximate solutions at previous x-steps to find the solution at the current step. Why use a multi-step method? Can get higher order truncation errors and are generally efficient since the computations are usually elementary. Again, there are implicit and explicit multistep methods.

Take initial value problem and advance one step (again, consideration is limited to equally-spaced nodes in x). Multi-step methods can be written as

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(t, y(t))dt = y_{n-r} + \int_{x_{n-r}}^{x_{n+1}} f(t, y(t))dt. \tag{42}$$

ADAMS-BASHFORTH FORMULA (AB)

$$\begin{gathered} y_{n+1} = y_n + af_n + bf_{n-1} + cf_{n-2} \cdots \\ \text{where } f_n \equiv f(x_n, y_n), y_n = y(x_n) \\ a, b, c, \ldots \text{ are constants} \end{gathered}$$

Example: Adams-Bashforth, order 5:

$$AB5 : y_{n+1} = y_n + \frac{h}{720}[1901f_n + 2616f_{n-2} + 251f_{n-4} - 2774f_{n-1} - 1274f_{n-3}]$$

Coefficients come from the following:

$$\int_{x_n}^{x_{n+1}} f(t, y(t))dt \approx h[Af_n + Bf_{n-1} + Cf_{n-2} + Df_{n-3} + Ef_{n-4}] \tag{43}$$

and require that (43) be exact when f is a polynomial of degree at most 4. Let us consider how the AB5 is constructed: Denote $\mathcal{P}_k$ be the family of polynomials of degree at most k. Recast problem of finding coefficients into a linear algebra problem . . . Without loss of generality, work this out at $x_n = 0$ and assume that $h = 1$.

$\mathcal{P}_4 = \{1, x, x(x+1), x(x+1)(x+2), x(x+1)(x+2)(x+3)\} \equiv \{p_0, p_1, p_2, p_3, p_4\}$ is a basis

then enforce

$$\int_0^1 p_n(t)dt = Ap_n(0) + Bp_n(-1) + Cp_n(-2) + Dp_n(-3) + Ep_n(-4)$$

obtain:

$$\begin{cases} A + B + C + D + E = 1 \\ -B - 2C - 3D - 4E = \frac{1}{2} \\ 2C + 6D + 12E = \frac{5}{6} \\ -6D - 24E = \frac{9}{4} \\ 24E = \frac{251}{30} \end{cases}$$

This is a "Method of Constant Coefficients", a general technique that can be used to obtain any order formula (see 475A notes on quadrature techniques).

Remark. One can generate $\mathcal{P}_n$ basis conveniently using Newton difference formulas (see 475A notes on Newton difference formulas). In fact, it is a good idea to review notes on quadrature and on interpolation, in particular, Gaussian and Chebychev Quadrature and interpolation, to draw conclusions on whether it makes sense to use a nonuniform grid from a practical point of view.

Exercise) Why are these not used in initial value problems all that often, if at all?

ADAMS-MOULTON FORMULAS (AM)

Assume (42) can be written as

$$y_{n+1} = y_n + af_{n+1} + bf_n + cf_{n-1} \cdots$$
$$\nearrow$$
new

Example: (AM5)

$$(44) \qquad y_{n+1} = y_n + \frac{h}{720}[251f_{n+1} + 646f_n + 106f_{n-2} - 264f_{n-1} - 19f_{n-3}]$$

derived by Method of Undetermined Coefficients (exercise).

Note the appearance of f_{n+1}, making this method implicit.

How to advance (44) in n?

Answer: use $AB5$ as "predictor" then an $AM5$ "corrector":

$$\begin{array}{ll} ABN & \tilde{y}_{n+1} = y_n + af_n + bf_{n-1} \cdots \quad \text{Predictor} \\ AMN & y_{n+1} = y_n + p\tilde{f}_{n+1} + qf_n \ldots \quad \text{Corrector} \end{array}$$

where N is the order (want to use same order for predictor and corrector, usually)

$$\text{and } \begin{cases} f_n = f(x_n, y_n) \\ \tilde{f}_n = f(x_n, \tilde{y}_n) \end{cases}$$

How to start integration?

Commonly $\rightarrow$ use ERKN (explicit Runge-Kutta of order N) to find enough y_n's for the multi-step to take over.

Another way $\rightarrow$ use incremental-order incremental-sized-step multi-step method.

Another way $\rightarrow$ use above in conjunction with fixed point iteration to find the implicit part of the AM stage.

Exercise) write down in detail the algorithmic strategies to accomplish these last two choices above.

Multi-Step Scheme Convergence and Stability

Assume $x_n = x_0 + nh$

$$\begin{gathered} n = 0, 1, 2 \ldots N(h) \\ b = x_0 + N(h)h \\ h = \frac{b - x_0}{N(h)}. \end{gathered}$$

As usual, let $y_n = y(x_n)$.

Express multi-step method as

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h \sum_{j=-1}^{p} b_j f(x_{n-j}, y_{n-j}) \tag{45}$$

$$x_{p+1} \leq x_{n+1} \leq b$$

For the initial value problem

$$\begin{cases} Y' = f(x, Y) \\ Y(0) = Y_0 \end{cases}$$

with $Y = Y(x)$, f continuous. Also assume there exists a Lipschitz constant.

Some definitions: we say

- **Stable Numerical Method** $\rightarrow$ if all the approximating solutions $\{y_n\}$ are "stable." Here $\{y_n | 0 \leq n \leq N(h)\}$ is an approximate solution.
- **Stable Approximating Solutions**: Let $\{y_n\}$ be approximate solutions for $h < h_0$ sufficiently small. For each $h \leq h_0$, perturb the initial values

$$y_0, y_1 \cdots y_p \text{ to new values } z_0, z_1 \ldots z_p \text{ with}$$
$$\max_{0 \leq n \leq p} |y_n - z_n| \leq \varepsilon \quad 0 < h \leq h_0$$

  Note: initial values usually depend on h.

  We say $\{y_n\}$ "stable" if $\exists\, c$, constant, independent of $h \leq h_0$, and valid for all ε small enough, for which

$$\max_{0 \leq h \leq N(h)} |y_n - z_n| \leq c\varepsilon \quad 0 < h \leq h_0.$$

□

- **Convergent Scheme** $\rightarrow$ for initial value problem: suppose initial values

$$y_0, y_1 \cdots y_p$$

  satisfy

$$\eta(h) \equiv \max_{0 \leq n \leq p} |Y(x_n) - y_n| \rightarrow 0 \text{ as } h \rightarrow 0$$

  $\Rightarrow \{y_n\}$ is said to converge to $Y(x)$ if

$$\max_{x_0 \leq x_n \leq b} |Y(x_n) - y_n| \rightarrow 0 \text{ as } h \rightarrow 0$$

  if (45) convergent for all y solutions of the IVP $\Rightarrow$ "Convergent Numerical Method".

Remark. Non-convergent numerical methods are useless in a practical setting!

□

- **Consistent Scheme**: if

$$\frac{1}{h}\max_{x_0\le x_n\le b}|T_n(Y)|\to 0 \text{ as } h\to 0 \ \forall Y(x) \text{ continuously differentiable on } [x_0,b]$$

$$\text{where } T_n(Y)\equiv \text{ truncation error}$$

$$\text{i.e. } T_n(Y)=Y(x_{n+1})\text{-numerical scheme}\Big|_{x_{n+1}}$$ □

Theorem. Convergence implies consistency.

Proof. (will be proven in context of numerical solution of partial differential equations, later discussed in this class) (see 0.4). □

Stability of Multi-step Methods
that is, of

$$y_{n+1}-\sum_{j=0}^{p}a_jy_{n-j}-h\sum_{j=-1}^{p}b_jf(x_{n-j},y_{n-j})=0.$$

which is a difference equation. We want to consider in detail the issue of stability of multi-step methods. Recall from our consideration of Difference Equation solutions to the Null-space problem (see See Difference Equations.) that it is sensible to assume that there exists a polynomial associated with (45) of the form

$$\rho(r)=r^{p+1}-\sum_{j=0}^{p}a_jr^{p-j}$$

Note: $\rho(1)=0$ from consistency condition:

$$\sum_{j=0}^{p}a_j=1 \qquad \text{(Theorem not quoted}$$
$$\sum_{j=0}^{p}ja_j+\sum_{j=-1}^{p}b_j=1 \qquad \begin{matrix}\text{shows that}\\ \leftarrow \\ \text{implies consistency)}\end{matrix}$$

Let $r_0=1,r_1,r_2\dots r_p$ be roots. Then

(45) satisfies "Root Condition" if

$$\begin{array}{ll} & |r_j| \le 1 \qquad j = 0, 1, \ldots p \\ \text{for} & |r_j| = 1 \Rightarrow \text{ these must be simple roots.} \end{array}$$

THE BIG PICTURE:

$$\begin{array}{ccccc} & & \text{Strong} & \Rightarrow & \text{Relative Stability} \\ & & \text{Root} & & \\ & & \Downarrow & & \Downarrow \\ \text{Convergence} & \Longleftrightarrow & \text{Root} & \Leftarrow & \text{Stability} \\ & & \text{Conditions} & & \end{array}$$

Theorem. (Stability) Suppose (45) is consistent. Then (45) is stable if and only if the root condition is satisfied.

Example:

$$y_{n+1} = 3y_n - 2y_{n-1} + \frac{h}{2}[f(x_n, y_n) - 3f(x_{n-1}, y_{n-1})] \quad n \ge 1$$

$$T_n(Y) = \frac{7}{12}h^3 Y'''(\xi_n) \qquad x_{n-1} \le \xi_n \le x_{n+1}$$

$$\text{consider} \begin{cases} y' = 0 \\ y(0) = 0 \end{cases} \qquad \Rightarrow Y(x) = 0$$

So if $y_0 = y_1 = 0$ and $y_n = 0 \;\; n \ge 0$. Perturbation $z_0 = \varepsilon/2, z_1 = \varepsilon \Rightarrow$

$$z_n = \varepsilon 2^{n-1} \quad n \ge 0$$

So

$$\max_{x_0 \le x_n \le b} |y_n - z_n| = \max_{0 \le x_n \le b} |\varepsilon| 2^{n-1} = |\varepsilon| 2^{N(h)-1}$$

$$\text{and } N(h) \to \infty \text{ as } h \to 0 \therefore \underline{\underline{\text{unstable}}}$$

Compute $\rho(r) = r^2 - 3r + 2$ with roots $r_0 = 1, r_1 = 2 \therefore$ Root condition violated. Since we're at it, we should probably also do the more general problem of looking at a system

$$\begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}) & \mathbf{y} \in \mathcal{R}^m \\ \mathbf{y}(0) = \mathbf{y}_0 & \mathbf{f} \in \mathcal{R}^m \end{cases}$$

if f is differentiable $\Rightarrow J \equiv \frac{\partial f_i}{\partial y_j} \quad 1 \le i, j \le m$

$$\Rightarrow \quad \mathbf{y}' = \wedge \mathbf{y} + \mathbf{g}(x) \qquad m \times m \text{ system}$$
$$\text{with } \wedge = f_y(\mathbf{x}_0, \mathbf{Y}_0) \qquad \text{with } \lambda_1, \lambda_2, \ldots \lambda_m \text{ eigenvalues.}$$

We can can make some headway in understanding what happens in the system case by considering what happens in the simpler problem

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases}$$

Using (45)

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h\lambda \sum_{j=-1}^{p} b_j y_{n-j}$$
$$(1 - h\lambda b_{-1}) y_{n+1} - \sum_{j=0}^{p} (a_j + h\lambda b_j) y_{n-j} = 0 \quad n \ge p$$

"homogeneous linear differential equation of order $p+1$"

See Difference Equations.

To solve, let

$$y_n = r^n \quad n \ge 0$$

and hope to find $p+1$ linearly independent solutions so that any solution can be expressed as a linear combination.

Substitute r^n and cancel r^{n-p}

$$(1 - h\lambda b_{-1}) r^{p+1} - \sum_{j=0}^{p} (a_j + h\lambda b_j) r^{p-j} = 0. \tag{46}$$

$$\text{Let } \sigma(r) \equiv b_{-1} r^{p+1} + \sum_{j=0}^{p} b_j r^{p-j},$$

(45) becomes

$$\rho(r) - h\lambda\sigma(r) = 0 \tag{47}$$

known as the "characteristic equation."

Denote roots as $r_0(h\lambda), \ldots r_p(h\lambda)$ depending continuously on $h\lambda$. When $h\lambda = 0$ (47) becomes

$$\rho(r) = 0 \qquad \text{so} \qquad r_j(0) = r_j \quad j = 0, 1 \ldots p.$$
$$\nearrow$$
$$\text{roots of } \rho(r).$$

If $r_j(h\lambda)$ are distinct then

$$y_n = \sum_{j=0}^{p} \gamma_j [r_j(h\lambda)]^n \quad n \geq 0$$

if $r_j(h\lambda)$ has a root of multiplicity $\nu > 1$ construct extra ν linearly independent by

$$[r_j(h\lambda)]^n, n[r_j(h\lambda)]^n, \ldots n^{\nu-1}[r_j(h\lambda)]^n \ldots$$

□

Proof. (Stability Theorem)

Here we present a simplified proof (see Isaacson and Keller '66 for full proof).

1) Assume root condition satisfied.

2) Roots are distinct $r_j(0)$ and $r_j(h\lambda) \quad 0 < h \leq h_0$.

Take z_n and y_n as solutions to

$$(1 - h\lambda b_{-1})y_{n+1} - \sum_{j=0}^{p}(a_j + h\lambda b_j)y_{n-j} = 0 \quad \text{on } [x_0, b]$$

let $e_n = y_n - z_n$ and assume

$$\max_{0 \leq n \leq p} |y_n - z_n| \leq \varepsilon \quad 0 \leq h \leq h_0 \tag{48}$$

$$(49) \quad \therefore e_{n+1}(1-h\lambda b_{-1}) - \sum_{j=0}^{p}(a_j + h\lambda b_j)e_{n-j} = 0 \quad \text{for } x_{p+1} \le x_{n+1} \le b$$

$$\text{with solution } e_n = \sum_{j=0}^{p} \gamma_j [r_j(h\lambda)]^n \quad n \ge 0$$

The coefficients must satisfy

$$\begin{bmatrix} \gamma_0 + \gamma_1 \cdots \gamma_p = e_0 \\ \gamma_0 r_0(h\lambda) + \cdots \gamma_p r_p(h\lambda) = e_1 \\ \vdots \\ \gamma_0 [r_0(h\lambda)]^p + \cdots \gamma_p [\gamma_P(h\lambda)]^p = e_p \end{bmatrix}$$

then $e_0 \dots e_p$ will satisfy (49)

Using linear theory and (48) $\Rightarrow \max_{0 \le i \le p} |\gamma_i| \le c_1 \varepsilon \;\; 0 < h \le h_0$

To bound e_n on $[x_0, b]$ we must bound each $[r_j(h\lambda)]^n$. To do so, consider

$$(50) \qquad r_j(u) = \gamma_j(0) + u r_j'(\zeta)$$

for some ζ between 0 and u (variation of parameters). Compute r_j': differentiate characteristic equation

$$\rho(r_j(u)) - u\sigma(r_j(u)) = 0$$

Therefore

$$(51) \qquad r_j'(u) = \frac{\sigma(r_j(u))}{\rho'(r_j(u)) - u\sigma'(r_j(u))}$$

by assumption $r_j(0)$ simple root of $\rho(r) = 0 \quad 0 \le j \le p \therefore \rho'(r_j(0)) \ne 0$ and by continuity, $\rho'(r_j(u)) \ne 0$ for all u small $\therefore$ denominator not zero and

$$|r_j'(u)| \le c_2 \qquad \forall |u| \le u_0 \qquad \text{for some } u_0 > 0.$$

Using (50) and the root condition: $|r_j| \le 1$, we get

$$|r_j(h\lambda)| \le |r_j(0)| + c_2|h\lambda| \le 1 + c_2|h\lambda|$$

$$|[r_j(h\lambda)]^n| \le [1 + c_2|h\lambda|]^n \le e^{c_2 n|h\lambda|} \le e^{c_2(b-x_0)|\lambda|} \forall h \le h_0.$$

$$\therefore \max_{x_0 \le x \le b} |e_n| \le c_3|\varepsilon| e^{c_2(b-x_0)|\lambda|} \quad 0 < h \le h_0$$

□

Theorem. (Convergence, Dahlquist Equivalence Theorem) Assume scheme is consistent. Then (45) is convergent if and only if root condition is satisfied.

Proof. Assume root condition is satisfied. Again, general proof in Isaacson and Keller. Assume $r_j(0)$ distinct.

Again

$$\begin{cases} y' = \lambda y \\ y(x_0) = 1 \end{cases}$$

and

$$\gamma_0[r_0(h\lambda)]^n$$

of

$$y_n = \sum_{j=0}^{p} \gamma_j [r_j(h\lambda)]^n$$

converges to solution $Y(x) = e^{\lambda x}$ on $[x_0, b]$. The remaining terms $\gamma_j[r_j(h\lambda)]^n j = 1, \ldots p$ are parasitic and shown to $\to 0$ as $h \to 0$.

Expand $r_0(h\lambda) = r_0(0) + h\lambda r'(0) + \mathcal{O}(h^2)$.

$$\text{From (51)} \qquad r_0(0) = \frac{\sigma(1)}{\rho'(1)}$$

and using consistency condition $\sum_{j=0}^{p} a_j = 1$ and $\sum_{j=0}^{p} j a_j + \sum_{j=-1}^{p} b_j = 1$ leads to $r_0'(0) = 1$. Then $r_0(h\lambda) = 1 + h\lambda + \mathcal{O}(h^2) = e^{\lambda h} + \mathcal{O}(h^2)$

$$[r_0(h\lambda)]^n = e^{\lambda n h}[1 + \mathcal{O}(h^2)]^n = e^{\lambda x_n}[1 + \mathcal{O}(h)]$$

over $x_0 \le x_n \le b$ finite.

Thus

$$\max_{0 \le x_n \le h} [|r_0(h\lambda)|^n - e^{\lambda x_n}] \to 0 \text{ as } h \to 0$$

We must now show that the coefficient $\gamma_0 \to 1$ as $h \to 0$. Again $\gamma_0(h) \cdots \gamma_p(h)$

satisfy

$$\left[\begin{array}{c} \gamma_0 + \cdots \gamma_p = y_0 \\ \gamma_0[r_0(h\lambda)] + \cdots \gamma_p[r_0(h\lambda)] = y, \\ \vdots \\ \gamma_0[r_0(h\lambda)]^P + \cdots \gamma_p[r_p(h\lambda)]^p = y_p \end{array}\right. \tag{52}$$

the initial values $y_0 \cdots y_p$ depend on h and satisfy

$$\begin{aligned} & \eta(h) \equiv \max_{0\le n\le p} |e^{\lambda x_n} - y_n| \to \text{ as } h \to 0 \\ \Rightarrow \quad & \lim_{h\to 0} y_n = 1 \quad 0 \le n \le p \end{aligned}$$

The coefficient $\gamma_0 \to 1$ as $h \to 0$ (look at solution of linear system (52) and see that by Cramer's the denominator converges to Vandermonde determinant for $r_0(0) = 1, r_1(0), \ldots r_3(0)$ nonzero and distinct roots. Same for numerator.

$$\therefore \{y_n\} \to e^{\lambda x} \text{ as } h \to 0 \text{ on } [x_0, b].$$

□

Corollary. If (42) consistent. Then convergent if and only if stable.

Proof. Follows directly from above theorems.

□

Relative and Weak Stability:

Consider

$$\left\{\begin{array}{l} y' = \lambda y \\ y(0) = 1 \end{array}\right.$$

and the general solution $y_n = \sum_{j=0}^{p} \gamma_j[r_j(h\lambda)]^n, \quad n \ge 0.$

The convergence theorem states that parasitic solutions of $\gamma_j[r_j(h\lambda)]^n \to 0$ as $h \to 0$. However, we use finite h and want, for any x_n, for them to be small compared to $\gamma_0[r_0(h\lambda)]^n$.

$$\text{need } \therefore \Rightarrow |r_j(h\lambda)| \le |r_0(h\lambda)| \quad j = 1, 2, \ldots p \tag{53}$$
$$\text{for } h \text{ sufficiently small.}$$

This leads to concept of "relative stability."

A method is "relatively stable" if the characteristic roots $r_j(h\lambda)$ satisfy (53) for all sufficiently small $|\lambda h|$. And a method satisfies the "strong root condition" if

$$|r_j(0)| < 1 \quad \text{for } j = 1, 2, \ldots p$$

Easy to check and it implies "relative stability."

Remark. Relative Stability does not imply the strong root condition (although they're equivalent for most methods). If multi-step method is stable but not relatively stable, it is "weakly stable."

Example: Using the bf midpoint method defined as $y_{n+1} = y_{n-1} + 2hf(x_n, y_n), n \geq 1$ to solve

$$\text{and } \begin{cases} Y' = \lambda Y \\ Y(0) = 1 \end{cases}$$

with exact solution

$$Y(x) = e^{\lambda x}.$$

Take $y_n = r^n \quad n \geq 0$

$$\begin{aligned} r^{n+1} &= r^{n-1} + 2h\lambda r^n \qquad \Rightarrow r^2 = 1 + 2h\lambda r \\ r_0 &= h\lambda + \sqrt{1 + h^2\lambda^2} \qquad r_1 = h\lambda - \sqrt{1 + h^2\lambda^2} \end{aligned}$$

so general solution

$$y_n = \beta_0 r_0^n + \beta_1 r_1^n, \quad n \geq 0 \tag{54}$$

$$\begin{cases} \beta_0 + \beta_1 = y_0 \\ \beta_0 r_0 + \beta_1 r_1 = y_1 \end{cases}$$

$$\therefore \beta_0 = \frac{y_1 - r_1 y_0}{r_0 - r_1} \qquad \beta_1 = \frac{y_0 r_0 - y_1}{r_0 - r_1}, \text{ generally}$$

using initial condition as above $\Rightarrow y_0 = 1, y_1 = e^{\lambda h}$

$$\begin{aligned} \beta_0 &= \frac{e^{\lambda h} - r_1}{2\sqrt{1 + h^2\lambda^2}} = 1 + \mathcal{O}(h^2\lambda^2) \\ \beta_1 &= \frac{r_0 - e^{\lambda h}}{2\sqrt{1 + \lambda^2 h^2}} = \mathcal{O}(h^3\lambda^3) \end{aligned}$$

$\beta_0 \to 1$ $\beta_1 \to 0$ as $h \to 0$ $\therefore$ $\beta_0 r_0^n$ in (54) should correspond to true solution $e^{\lambda x_n}$, since $\beta_1 r_1^n \to 0$ as $h \to 0$. In fact

$$r_0^n = e^{\lambda x_n}[1 + \mathcal{O}(h)]$$

Now, assume λ is real and positive (for illustration)

$$\text{then } r_0 > |r_1| > 0$$

thus r_1^n increases less rapidly than r_0^n so $\beta_0 r_0^n$ will dominate. Now, assume λ is real and negative

$$\text{then } 0 < r_0 < 1 \quad r_1 < -1 \quad h > 0$$

$\therefore$ $\beta_1 r_1^n$ will dominate $\beta_0 r_0^n$ as $n \to \infty$, for fixed h, no matter how small h. The $\beta_0 r_0^n \to 0$ as $n \to \infty$, whereas the term $\beta_1 r_1^n$ increases, alternating sign as $n \to \infty$.

The $\beta_1 r_1^n$ is the "parasitic" solution (a creation of the numerical method) $\Rightarrow$ Midpoint method is "weakly stable" ... the parasitic solution will eventually make the solution diverge from the solution.

In summary, the midpoint method is weakly stable according to (53) since

$$r_0(h\lambda) = 1 + h\lambda + \mathcal{O}(h^2) \quad r_1(h\lambda) = -1 + h\lambda + \mathcal{O}(h^2)$$

for $\lambda < 0$.

Example: Try AB and AM. They have same characteristic polynomial when $h = 0$:

$$\rho(r) = r^{p+1} - r^p$$

The roots are $r_0 = 1, r_j = 0 \quad j = 1, \ldots p$ $\therefore$ Strong Root condition is satisfied and Adams methods are relatively stable.

□

0.1.11 Backward Differentiation Formulas (BDF's)

Multi-step methods built with superior stability properties.

Construction: $P_p(x) \equiv$ polynomial of degree $\leq p$ that interpolates $Y(x)$ at $x_{n+1}, x_n, \ldots x_{n-p+1}$ for some $p \geq 1$:

$$P_p(x) = \sum_{j=-1}^{p-1} Y(x_{n-j})\ell_{j,n}(x) \tag{55}$$

where

$$\ell_{j,n}(x) = \prod_{\substack{j=n+1 \\ j\neq 1}}^{n-p+1} \frac{x - x_j}{x_i - x_j} \qquad n+1 \leq i \leq n-p+1$$

Lagrange interpolation basis functions, for nodes $x_{n+1} \ldots x_{n-p+1}$ differentiate

$$P_p'(x_{n+1}) \approx Y'(x_{n+1}) = f(x_{n+1}, Y(x_{n+1})) \tag{56}$$

Combine (55) and (56):

$$Y_{n+1} \approx \sum_{j=0}^{p-1} \alpha_j Y(x_{n-j}) + h\beta f(x_{n+1}, Y_{n+1})$$

$\therefore$ p-step method is

$$y_{n+1} = \sum_{j=0}^{p-1} \alpha_j y_{n-j} + h\beta f(x_{n+1}, y_{n+1})$$

Can find coefficients α_j, β for given p in many numerical analysis books.

Truncation error for the method:

$$T_n(Y) = -\frac{\beta}{p+1} h^{p+1} Y^{(p+1)}(\xi_n) x_{n-p+1} \leq \xi_n \leq x_{n+1}$$

Exercise: derive truncation formula (hint, review Lagrange Interpolation).

Example:

$p = 1 \quad \beta = 1 \quad \alpha_0 = 1$ get Implicit Euler

$p = 2 \quad y_{n+2} - \frac{4}{3} y_{n+1} + \frac{1}{3} y_n = \frac{2}{3} h f(x_{n+2} Y_{n+2})$

$p = 3 \quad y_{n+3} - \frac{18}{11} y_{n+2} + \frac{9}{11} y_{n+1} - \frac{2}{11} y_n = \frac{6}{11} h f(x_{n+3}, y_{n+3})$

□

0.1.12 Stability and Stiff Equations

Perhaps the best way to motivate this concept is by looking at an example (taken from Iserles). Let

$$A = \begin{bmatrix} -20 & 10 & 0 & \cdot & \cdot & \cdot & 0 \\ 10 & -20 & +10 & & & & : \\ 0 & \cdot & \cdot & \cdot & & & : \\ : & & \cdot & & & & : \\ : & & \cdot & \cdot & \cdot & & : \\ : & & \cdot & \cdot & \cdot & & : \\ : & & & 10 & -20 & 10 & 0 \\ : & & & & 10 & -20 & 10 \\ 0 & \cdot & \cdot & \cdot & 0 & 10 & -20 \end{bmatrix}$$

We solve

$$\begin{aligned} \mathbf{Y}' &= A\mathbf{Y} \\ \mathbf{Y}(0) &= \mathbf{I}, \end{aligned} \tag{57}$$

using AB2. To be specific, Figure (7) shows the Sup-norm of the approximate solution $y(t)$, for $M = 10$, the size of the $M \times M$ matrix $\mathbf{A}$, for two and slightly different values of h, the step size. The dashed line corresponds to the approximate solution with $h = 2.702703 \times 10^{-2}$ and the solid to $h = 2.73972 \times 10^{-2}$. You can download the matlab code that was used in this example. Notice from the plots that the approximates are drastically different even though the difference in the values of h is small. Why? after all, the difference in h is so small. If you were to try AB of 3^{rd} order you'd find it makes matters worse! If we had tried a BDF (low p, more on this later), the solution and the approximation would be reasonably close. □

Recall our analysis of Euler on the problem $y' = \lambda y$. One is tempted to conclude that a low-order method has poor approximating properties, for certain λ, h, as compared to a high order method.

But it is not the order of the method that caused the problem in the above example. Recall our analysis of trapezoidal scheme on $y' = \lambda y \rightarrow 2^{nd}$ order method that showed the correct asymptotic behavior IRRESPECTIVE of h!

In summary: we need to understand the distinction between the "order" of the method and its stability, e.g. The trapezoidal has superior stability

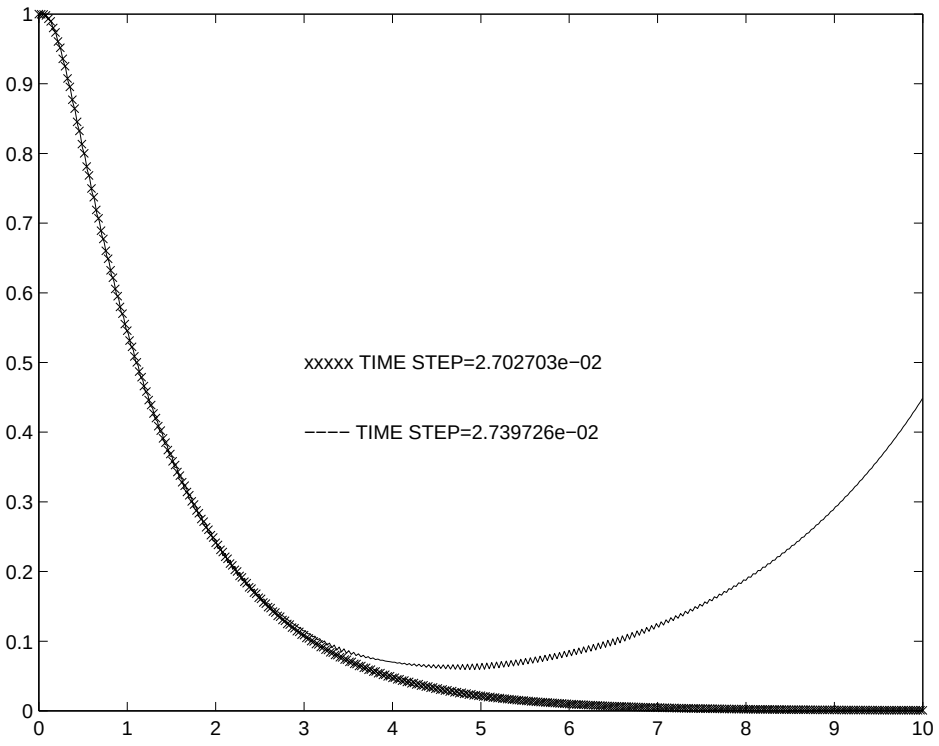

Figure 7: Sup-norm of approximate solution $y(t_n)$ of (57) with $M = 10$ using AB2. The dashed line corresponds to $h = 2.702703 \times 10^{-2}$ and the solid line to $h = 2.73972 \times 10^{-2}$

properties: in fact, we would find it to be stable independent of h! This does not mean that we can choose h arbitrarily and expect the approximation and the solution to be close to each other: convergent and stable are not the same as accurate. However, any scheme which is consistent, convergent, and stable will be accurate if we take h sufficiently small.

□

What's a stiff equation? No precise definition exists. Operationally,

$$\begin{cases} Y' = f(x, Y) \\ Y(x_0) = Y_0 \end{cases}$$

is "STIFF" if its numerical solution by some methods requires (perhaps in a portion of an interval) a significant depression of the step size in order to avoid instabilities.

Example: One way to assess qualitatively the stiffness of a system of equations is this: Take A a matrix of constant coefficients

$$\left\{ \begin{array}{l|l} \mathbf{Y}' = A\mathbf{Y} & \text{stiff if and only if} \\ \mathbf{Y}(x_0) = \mathbf{Y}_0 & Re(\lambda_j) < 0 \text{ and very large} \cdots \\ \text{with eigenvalues of } A \equiv \Lambda_j & \text{or ratio between largest and smallest eigenvalue is huge!} \end{array} \right.$$

Sometimes we see "Stiffness-ratio" as a way to "quantify" stiffness and is taken as ratio of the modules of largest to smallest eigenvalue of linearized system.

What's big? 10^3 and above, perhaps.

Example: Kinetic Reactions have coupled systems with stiffness ratio $\approx 10^{17}$

Example: Bigbang (Einstein's General Theory) stiffness ratio of $\approx 10^{31}$.

The Linear Stability domain and A-Stability

Remark

There are serious limitations to linear stability theory. Nevertheless, it is

very useful ... and EASY!

$$\begin{array}{lll} \text{Take} & Y' = \lambda Y & \lambda \text{ in } \mathbb{C}(\text{complex}) \\ & Y(0) = 1 & \\ \text{solution} & Y = e^{\lambda t} \quad \text{and } \lim_{t\to\infty} Y(t) = 0 & \text{if } Re(\lambda) < 0. \end{array}$$

We say that the "linear stability domain $\mathbb{D}$" of a numerical scheme is the $\{h\lambda\}$ set such that $\lim_{n\to\infty} y_n = 0$ with $h > 0$ and $\lambda \in \mathbb{C}$.

i.e. the set for which we obtain the correct asymptotic behavior.

Note: $Re(\lambda) > 0$ case used to be of limited interest ... solution grows rapidly and becomes very large. However, there's renewed interest in nonlinear problems ... there's a counterpart of λ called the "Liapunov" exponent ... (see Dynamical Systems text).

Example: Approximate the solution $Y(t)$ of

$$Y' = \lambda Y \qquad Y(0) = 1, \tag{58}$$

where $t \geq 0$, and λ is complex. Using Forward Euler, it is clear that the approximate solution is $y_n = (1 + h\lambda)^n$ where $n = 0, 1 \cdots$

$\therefore \{y_n\}$with $n = 0, 1 \ldots$ is a geometric sequence and $\lim_{n\to\infty} y_n = 0$ if and only if $|1 + h\lambda| < 1$

$$\therefore \mathbb{D}_{\mathrm{E}uler} = \{z \in \mathbb{C} : |1 + z| < 1\} \qquad z \equiv h\lambda$$

A domain which is wholy inscribed in a circle in the complex plane of z , centered at $(-1, 0)$, where the first entry corresponds to the real part of z.

Example: To illustrate the vector equation case, consider Forward Euler, applied to

$$\begin{cases} \mathbf{Y}' = \mathbf{\Lambda Y} \\ Y(0) = Y_0 \end{cases} \qquad \mathbf{\Lambda} = \begin{bmatrix} -100 & 1 \\ 0 & -\dfrac{1}{10} \end{bmatrix} \tag{59}$$

a simple vector case. Then

$$\mathbf{y}_1 = \mathbf{y}_0 + h\Lambda\mathbf{y}_0 = (1 + h\Lambda)\mathbf{y}_0 \quad y_2 = \mathbf{y}_1 + h\Lambda\mathbf{y}_1 = (1 + h\Lambda)^2\mathbf{y}_0$$

$$\therefore \mathbf{y}_n = (1 + h\Lambda)^n \mathbf{y}_0 \quad n = 0, 1 \cdots$$

Perform a spectral factorization

$$\mathbf{\Lambda} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \quad \mathbf{V} = \begin{bmatrix} 1 & 1 \\ 0 & \frac{999}{10} \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} -100 & 0 \\ 0 & -\frac{1}{10} \end{bmatrix}$$

$$\therefore \mathbf{Y}(t) = e^{\mathbf{\Lambda} x} = \mathbf{V} e^{x\mathbf{D}} \mathbf{V}^{-1} \mathbf{Y}_0, \quad x \geq 0$$

$$e^{x\mathbf{D}} = \begin{bmatrix} e^{-100x} & 0 \\ 0 & e^{-\frac{x}{10}}. \end{bmatrix}$$

$$\mathbf{Y}(t) = e^{-100x}\mathbf{S}_1 + e^{-\frac{x}{10}}\mathbf{S}_2 \quad x \geq 0$$

Euler approximate solution is thus

$$\mathbf{y}_n = \mathbf{V}(I + h\mathbf{D})^n \mathbf{V}^{-1} \mathbf{y}_0 \quad n = 0, 1 \cdots$$

with

$$(I + h\mathbf{D})^n = \begin{bmatrix} (1 - 100h)^n & 0 \\ 0 & \left(1 - \frac{1}{10}h\right)^n \end{bmatrix}$$

$$\therefore \mathbf{y}_n = (1 - 100h)^n \mathbf{S}_1 + \left(1 - \frac{1}{10}h\right)^n \mathbf{S}_2 \tag{60}$$

□

Consider now the $d \times d$ generalization of the problem considered in (59):

$$\begin{aligned} \mathbf{D} &= \text{diag}\{\lambda_1, \cdots \lambda_3\} \text{and } \mathbf{S_1}, \mathbf{S_2}, \cdots \mathbf{S_d} \in \mathbb{C} \text{ depending on} \mathbf{Y}_0. \\ \mathbf{y}_n &= \sum_{k=1}^{d} (1 + h\lambda_k)^n \mathbf{S}_k \quad n = 0, 1 \cdots \end{aligned}$$

so the requirement that $|1 + b\Lambda_k| < 1$, for $k = 1, 2 \cdots d$ means that all $h\lambda_1, h\lambda_2, \cdots h\lambda_k$ lie in $\mathbb{D}_{\mathrm{E}_{uler}}$.

Remark In practice, sufficient to look at stiffest component: THE STEP SIZE IS CONTROLLED BY STIFFEST COMPONENT.
The above example assumed there was a full set of eigenvectors. Generally,

can use Jordan Factorization (since every $d \times d$ possesses such factorization) to find full set to illustrate, assume $\mathbf{\Lambda}$ is $d \times d$. Let $\mathbf{\Lambda} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}$

$\det \mathbf{W} \neq 0$

so that $\mathbf{\Lambda} = \begin{bmatrix} \Lambda_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \Lambda_{S-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \Lambda_S \end{bmatrix}$

there $\lambda_1, \lambda_2\lambda_3 \cdots \in \sigma(A)$ and the k^{th} Jordan-block is

$$\Lambda_k = \begin{bmatrix} \lambda_k & 1 & 0 & \cdots & 0 \\ 0 & \lambda_k & 1 & \ddots & : \\ : & \ddots & \ddots & \ddots & 0 \\ : & \ddots & 0 & \lambda_k & 1 \\ 0 & 0 & 0 & 0 & \lambda_k \end{bmatrix} k = 1, 2, \cdots s.$$

□

Remark

Example: Trapezoidal (bearing in mind that $Y_0 = 1$ at $x = 0$)

$$y_n = \left(\frac{1 + \frac{1}{2}h\lambda}{1 - \frac{1}{2}h\lambda}\right)^n \quad n = 0, 1$$

$$\mathbb{D}_{TRAP} = \left\{ z \in \mathbb{C} : \left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| < 1 \right\}$$

Thus the region of stability is the whole left-hand plane of z such that $Re(z) < 0$. In fact, since it is a useful concept, let's denote

$$\mathbb{C}^- \equiv \{ z \in \mathbb{C} : Re(z) < 0 \}$$

Since trap mimics asymptotic stability of linear ODE without limitations on h, we define this as "A-STABLE."

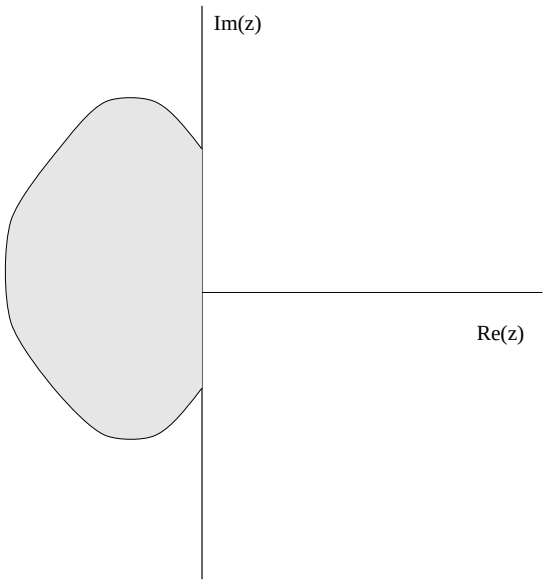

Figure 8: $\hat{r}_{3/0}$. Stability region is bounded region.

define: Method is "A-Stable" if

$$\mathbb{C}^- \subseteq \mathbb{D}$$

Hence, A-stable methods enable you to choose the step-size h based on accuracy considerations, rather than on stability considerations.

Exercise: Verify that the Theta Method is A-Stable if and only if $\frac{1}{2} \leq \theta \leq 1$. $\square$

definition: Let $\hat{r}_{1/1} \equiv \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$. We say that this ratio of polynomials in z is the Padé approximant $\hat{r}_{1/1}$. Padé approximants will be discussed shortly.

Example: Take the Padé approximant $\hat{r}_{3/0} \equiv 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3$. If it was obtained from a numerical scheme for solving the initial value problem (58), the corresponding scheme would have stability in the bounded region of Figure 8. This scheme would be an example of a non A-stable method. Why?
Example: Take the Padé approximant

$$\hat{r}_{1/2} \equiv \frac{1 + \frac{1}{3}z}{1 - \frac{2}{3}z + \frac{1}{6}z^2}.$$

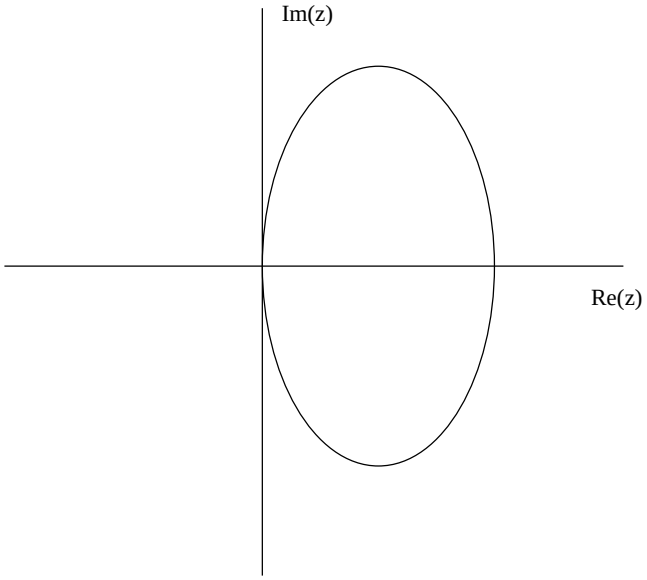

Figure 9: $\hat{r}_{1/2}$. Stability region is **outside of bounded region**.

If it was obtained from a numerical scheme for solving the initial value problem (58), the corresponding scheme would have a stability region **outside the bounded region** of Figure 9. This scheme would be an example of an A-stable method. Why? As you can see, the subscript of $\hat{r}$ corresponds to the order of the polynomial in the numerator and the denominator, respectively.

A-Stability of Runge-Kutta

Take

$$(61) \qquad \begin{cases} \xi_j = y_n + \sum_{i=1}^{\nu} a_{j,i} f(x_n + c_i h, \xi_i) \quad j = 1, 2, \cdots, \nu \\ y_{n+1} = y_n + h \sum_{j=1}^{\nu} b_j f(x_n + c_j h, \xi_j) \end{cases}$$

convention: $\sum_{i=1}^{\nu} a_{j,i} = c_j \qquad j = 1, 2, \cdots \nu$

Example:

2-Stage IRK

$$\begin{aligned}
\xi_1 &= y_n + \frac{1}{4}h\left[f(x_n,\xi_1) - f\left(x_n+\frac{2}{3}h,\xi_2\right)\right] \\
xi_2 &= y_n + \frac{1}{12}h\left[3f(x_n,\xi_1) + 5f\left(x_n+\frac{2}{3}h,\xi_2\right)\right] \\
y_{n+1} &= y_n + \frac{1}{4}h\left[f(x_n,\xi_1) + \xi f(x_n+\frac{2}{3}h,\xi_2)\right]
\end{aligned}$$

Tableaux:

0	$\frac{1}{4}$	$-\frac{1}{4}$
$\frac{2}{3}$	$\frac{1}{4}$	$\frac{5}{12}$
	$\frac{1}{4}$	$\frac{3}{4}$

□

Take (61) and use to approximate $\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases}$

$$\text{get } \xi_j = y_n + h\lambda \sum_{i=1}^{\nu} a_{j,i}\xi_i \quad j = 1, 2, \cdots, \nu$$

$$\text{let } \xi = (\xi_1, \xi_2, \cdots \xi_\nu)^T \text{and } \mathbf{1} = (1, 1, \cdots 1)^T \in \mathcal{R}^\nu$$

$$\text{then } \xi = \mathbf{1}y_n + h\lambda A\xi$$

$$\therefore \xi = (I - h\lambda A)^{-1}\mathbf{1}y_n$$

$$(62) \quad \therefore \quad y_{n+1} = y_n + h\sum_{j=1}^{\nu} b_j\xi_j = [1 + h\lambda \mathbf{b}^T(I - h\lambda A)^{-1}]y_n \quad n = 0, 1\cdots$$

Lemma

For every Runge-Kutta $\exists\, \hat{r} \in \mathbb{P}_{\nu/\nu}$ such that

$$(63) \qquad y_n[r(h\lambda)]^n \quad n = 0, 1\cdots$$

Moreover, for ERK $\Rightarrow \hat{r} \in \mathbb{P}$

Here, $\mathbb{P}_{\alpha/\beta}$ are rational functions $\hat{p}/\hat{q}$ such that $\hat{p} \in \mathbb{P}_\alpha, \hat{q} \in \mathbb{P}_\beta$.

Proof (outline)

So $\hat{r}(z) = 1 + z\mathbf{b}^T(I - zA)^{-1}\mathbf{1} \quad z \in \mathbb{C}$.

by (62) (63)

Need to show that $\hat{r}(z)$ is a rational function. Use

$$(I - zA)^{-1} = \frac{\text{adj}(I - zA)}{\det(I - zA)}$$

where *adj* is the "adjunct" of the matrix. The rest of proof omitted: must show that indeed, $\nu(z) \in \mathbb{P}_{\nu/\nu}$

□

Remark

A is strictly lower triangular if ERK $\therefore$ det $(I - zA) \equiv \mathbf{1}$ and $\hat{r}$ is a polynomial, rather than a ratio of polynomials, $\therefore$

Lemma The application of a numerical method to $y' = \lambda y$ that produces

$$y_n = [r(h\lambda)]^n \quad n = 0, 1 \cdots$$

where r is an arbitrary function. Then

$$\mathbb{D} = \{z \in \mathbb{C} : |r(z)| < 1|\}$$

Proof

Follows from definition of $\mathbb{D}$.

Corollary No ERK method is A-Stable

Proof

$r(z)$ is a polynomial and $r(0) = 1$.

□

Figure (10) shows some stability boundaries for ERK's of different orders.

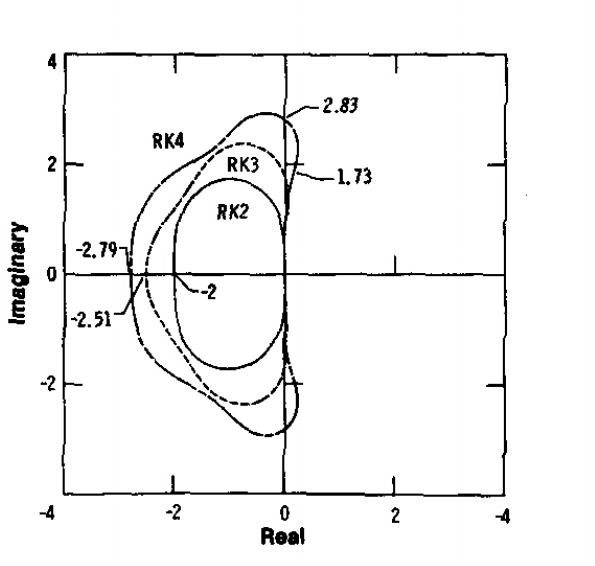

Figure 10: Stability regions for ERK of different orders.

Exercise: Verify that the 2 stage IRK above gives $\hat{r}_{\frac{1}{2}}(z)$ pictured in Figure 9.

Remark

What is $\hat{r}_{\alpha/\beta}$? These are Padé approximants, to the exponential e^z:

$$\hat{r}_{1/0} = 1 + z \quad \hat{r}_{1/1} = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \quad \hat{r}_{1/2}(z) = \frac{1 + \frac{1}{3}z}{1 - \frac{2}{3}z + \frac{1}{6}z^2}$$

Padé approximants will come up later when we study the stability of schemes for the solution of PDE's.

Example

Let's close this section with a couple of schemes we've seen before: Implicit Euler and Trapezoidal. In this example we verify the stability of these two methods, but in addition, point out a very interesting and sometimes very important aspect that distinguishes Euler and Trap.

Implicit Euler $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \quad n \geq 0$ apply to $y' = \lambda y$. Assume $Re(\lambda) < 0$.

$$y_n = \left[\frac{1}{1 - h\lambda}\right]^n y_0 \quad n \geq 0$$

then $y_n \to 0$ as $x_n \to \infty$ if and only if $|1 - h\lambda|^{-1} < 1$, true for all $h\lambda$ $\therefore$ A-Stable.

In fact for large magnitude $Re(\lambda) < 0, \quad y_n \to 0 \quad$ as $x_n \to \infty$ very quickly as it does for the exact solution $Y(x) = e^{\lambda x}$

THIS IS GOOD.

Compare this to A-STABLE Trap: $y_n = \left[\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right]^n y_0$

if $|Re(\lambda)|$ large, fraction inside $[\quad] \approx -1$ and y_n decreases very slowly!!

Remark If the problem is stiff, functional iteration, or fixed point methods will work (but must check) in the trapezoidal case. However, Use Newton rather than fixed point methods when solving Implicit Euler $\cdots$ Why? For stiff problems require $\left|h\frac{\partial f}{\partial y} < 1\right|$ forcing h to be tiny.

A-Stability of Multistep Methods

Multi-step methods such as Backward Differentiation Formulas (BDF's), AB, AM, and are considered next. What are BDF's? This is a family of multi-step implicit methods that have very good stability properties. Detailed consideration of them is beyond the scope of the course. They are classified by the number of stages p in the method. The $p = 1$ BDF, i.e. BDF1 is Implicit Euler. The next two are:

$$\begin{aligned} y_{n+2} &= \frac{4}{3}y_{n+1} - \frac{1}{3}y_n + \frac{2}{3}f(x_{n+2}, y_{n+2}) \\ y_{n+3} &= \frac{18}{11}y_{n+2} - \frac{9}{11}y_{n+1} + \frac{2}{11}y_n + \frac{6}{11}f(x_{n+3}, y_{n+3}) \end{aligned}$$

corresponding to BDF2 and BDF3, respectively.

Theorem

The root condition is satisfied if and only if $1 \le p \le 6$. Only then is the BDF schemes are convergent. □

Only $p = 1$, and $p = 2$, i.e. BDF1 and BDF2, are A-Stable. The BDF2 is

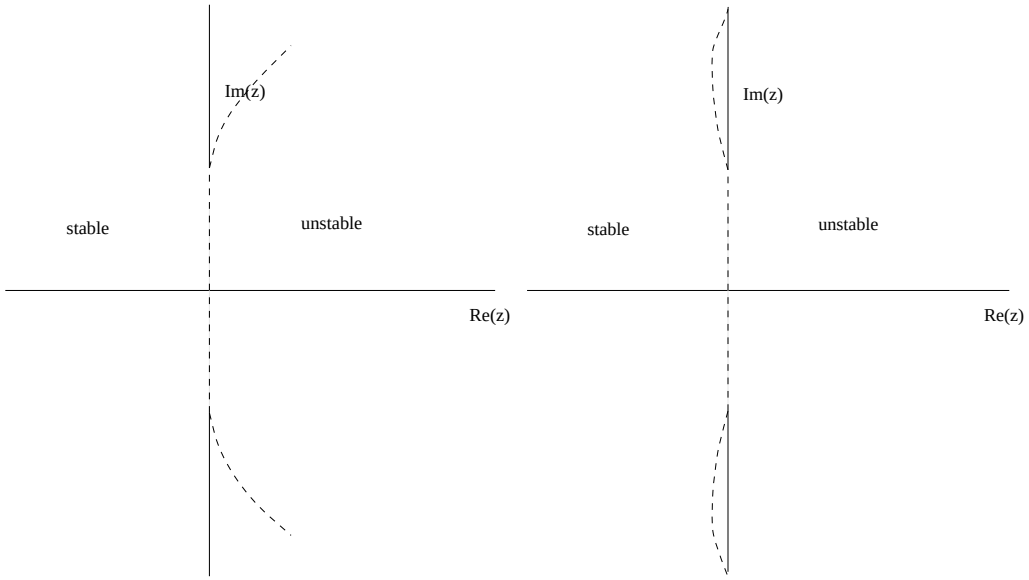

Figure 11: Approximate stability regions for BDF2, and BDF3. Stable regions are bounded to the right by the dashed curve.

actually very useful in problems that require stability when the eigenvalues λ of the matrix A, in the vector problem $y' = Ay$ has a slight $Re(\lambda) > 0$, but only slightly. This happens quite often in all sorts of applications, when for example, we want to stabilize a weakly unstable problem (see 0.5).

In fact one can show that the stability region for the BDF2 and BDF3 are approximately as pictured in Figure 11. In the figures, the stable region is bounded to the right by the dashed curve.

Theorem

(The Dahlquist-Second Barrier): The highest order of an A-Stable Multistep Method is 2.

Proof

See Lambert '91 Num. Methods for ODE's, Wiley.

□

Remark Suppose we want high-order and A-Stable? We'll have to resort to

IRK... yuk!!

Linear Stability of Adams-Bashforth and Adams-Moulton Schemes

In Figure (12), which is taken from "Spectral Methods" book from Hussaini, *et al*, we reproduce the stability boundaries for several AB and AM methods...

Remember that a p-step multistep method requires p values, including the initial condition, i.e. the "initial values." Since we only have 1 of these values, we must recast the stability issue in terms that are much stronger than is practically-necessary: We require linear stability FOR ALL POSSIBLE VALUES OF $y_0, y_1, \cdots y_{p-1}$:

Write multi-step method as

$$\sum_{m=0}^{P} a_m y_{n+m} = h\lambda \sum_{m=0}^{P} b_m y_{n+m} \quad n = 0, 1 \cdots \tag{64}$$

when applied to $y' = \lambda y$.

(64) written as

$$\sum_{m=0}^{P} (a_m - h\lambda b_m) y_{n+m} = 0 \quad n = 0, 1 \cdots \tag{65}$$

Get linear difference equation (see notes on linear-diff equations for a brush-up on topic).

To solve (64), form characteristic polynomial

$$c(w) \equiv \sum_{m=0}^{P} g_m w^m$$
$$g(m) = a_m - h\lambda b_m$$

Let $w_1, w_2, \cdots w_q$ be the zeros of $c(w)$ with multiplicatives $k_1, k_2, k_3, \cdots k_q$

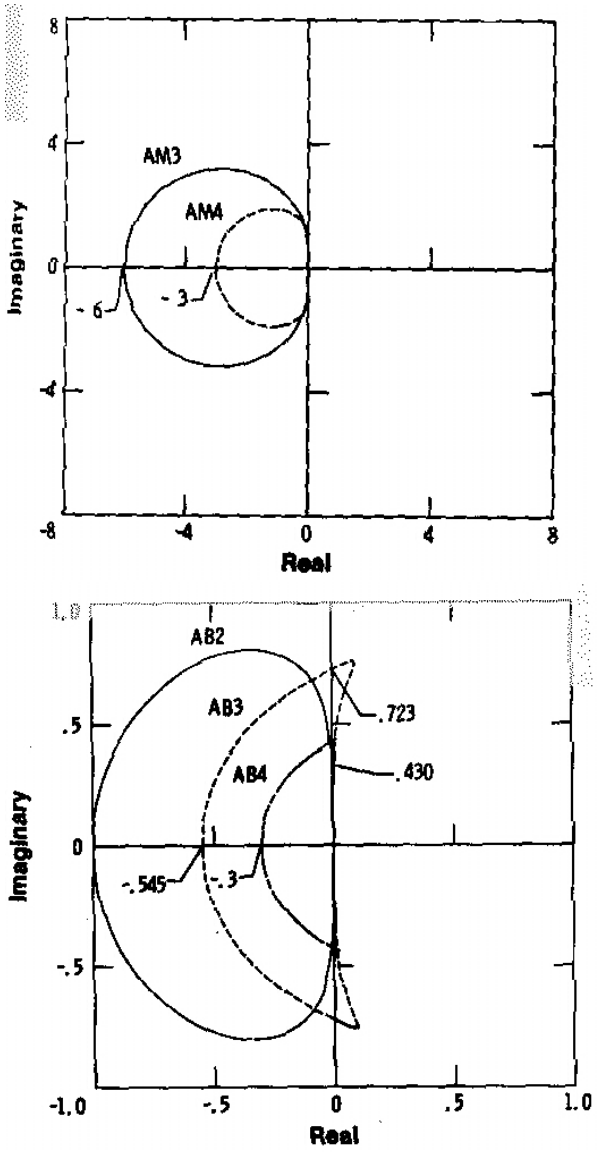

Figure 12: Stability Boundaries for several AM and AB schemes

where $\sum_{i=1}^{q} k_i = p$. Then the general solution of (64)

$$y_n = \sum_{i=1}^{q} \left(\sum_{j=0}^{k_i - 1} a_{i,j} n^j \right) w_i^n \quad n = 0, 1 \cdots \tag{66}$$

The constants are $p\, a_{i,j}$ uniquely determined by the p starting values $y_0, y_1, \cdots y_{p-1}$.

Lemma (A-Stability for Multi-Step): Suppose the zeros (as a function of w) of

$$c(z, w) = \sum_{m=0}^{P} (a_m - b_m z) w^m \quad z \in \mathbb{C} \quad z = \lambda h$$

are $w_1(z), w_2(z) \cdots, w_{q(z)}(z)$ and their multiplicatives $k_1(z), k_2(z) \cdots k_{q(z)}(z)$ respectively. Then the multi-step method (1) is A-stable if and only if

$$|w_i(z)| < 1 \quad i = 1, 2 \cdots q(z) \quad \forall z \in \mathbb{C} \tag{67}$$

Proof

Examining (65) we see that y_n behavior is determined by magnitude of $w_i(h\lambda)\, i = 1, 2 \cdots, q(h\lambda)$. If they all reside inside complex unit disk then their powers decay faster than any polynomial in n, thus, $y_n \to 0$

Hence (67) is sufficient for A-Stability.

On the other hand, if $|w_i(h\lambda)| \geq 1$, say, then there exist starting values such that $a_{1,0} \neq 0 \therefore$ it is impossible for $y_n \to 0$ as $x_n \to \infty$. We deduce that (66) is necessary for A-Stability.

□

Example

Is the AB $y_{n+1} = y_n + \frac{h\lambda}{2} [3yn - y_{n-1}]$ solution for $Y' = \lambda Y$ A-Stable?

$$, n \geq 1$$

The characteristic equation

$$r^2 - (1 + \frac{3}{2}h\lambda)r + \frac{1}{2}h\lambda = 0$$
$$r^2 - (1 + \frac{3}{2}z)r + \frac{1}{2}z = 0$$

The roots are: $r_{0,1} = \frac{1}{2}\left\{1 + \frac{3}{2}z \pm \sqrt{1 + z + \frac{9}{4}z^2}\right\}$

Region of absolute stability are such that $|r_0(z)| < 1, |r_1(z) < 1$ so $-1 < z < 0$. Thus not A-Stable.

General Comments comparing AB and BM:

1) Find that for both, region of absolute stability becomes smaller the higher the order.

2) For a given order, region of absolute stability is larger for AM.

3) Size of region usually acceptable from the point of view of practicality.

4) The Adams family is very easy to adapt to variable order (DEABM is a popular fortran code that does this).

5) No Adams scheme is A-Stable. Also, in general, the higher the order, the smaller the region of stability, but with higher order you get to include more of the right hand side of the eigenvalue plane. Hence, for mildly stiff problems and slightly unstable problems one can use a high order Adams, provided h is small enough.

0.2 BOUNDARY VALUE PROBLEMS (BVP)

We're not going to do justice to this topic, unfortunately. It is a vast subject, so we want to present some essential material that will give you a starting point for more advanced and related material. We will only concentrate on second order problems, for specificity.

Consider

$$(68) \qquad \begin{cases} Y'' = f(x, Y, Y') & x_0 \le x \le b \\ Y(x_0) = \alpha \\ Y(b) = \beta \end{cases}$$

$$\text{Some Methods for the Numerical Solution are} \begin{cases} \text{Shooting Methods} \\ \text{Finite Difference} \\ \text{Galerkin} \\ \text{Collocation} \\ \text{Rayleigh-Ritz} \end{cases}$$

We will study shooting and finite difference methods and limit ourselves to cursory comments on the other methods listed above.

First, we need a little bit of theory (see See Boyce and DiPrima book, for a nice presentation of this material..

Theorem (Existence and Uniqueness of Solutions)

Suppose f in equation (68) is continuous on $D = \left\{(x, Y, Y') \,|x_0 \le x \le b, -\infty < Y, Y' < \infty\right\}$

and that $\dfrac{\partial f}{\partial Y}$ and $\dfrac{\partial f}{\partial y'}$ are also continuous on D.

If

a) $\dfrac{\partial f}{\partial Y}(x, Y, Y') > 0 \quad \forall\, (x, Y, Y') \in D$ and

b) $\left|\dfrac{\partial f}{\partial Y'}(x, Y, Y')\right| \le M$, constant $\quad \forall\, (x, Y, Y') \in D$

$$\Rightarrow \text{BVP (68) has a unique solution}$$

□

Proof: Omitted. See Boyce and DiPrima book, for a nice presentation of this material. for details.

Corollary: If the following linear BVP

$$Y'' = p(x)Y' + q(x)Y + Y(x) \quad , \quad x_0 \le x \le b, \quad Y(x_0) = \alpha, \quad Y(b) = \beta$$

satisfies

i) $p(x), q(x), r(x)$ continuous on $[x_0, b]$

ii) $q(x) > 0$ on $[a, b]$ $\Rightarrow$ this BVP has a unique solution

□

Relation Between Homogenous and Inhomogeneous Problem

In order to understand the Shooting Method 0.2 it is important that we refresh your memory on the following issue: Take

$$Y'' = p(x)Y' + g(x)Y + r(x) \tag{69}$$

the inhomogeneous problem, with

$$\begin{aligned} Y(x_0) &= \alpha \\ Y(b) &= \beta \end{aligned}$$

under assumptions that ensure uniqueness will generate a solution $y^P(x)$, which we'll call the "particular solution."

Let $w(x) \equiv y(x) - y^P(x)$ where

$$\begin{aligned} w'' &= p(x)w' + q(x)w \\ w(x_0) &= \alpha - y(x_0) \equiv \alpha' \\ w(b) &= \beta - y^p(b) \equiv \beta' \end{aligned} \tag{70}$$

This is a homogeneous ODE with possibly inhomogeneous boundary conditions. $\therefore$ Can find a solution to inhomogeneous problem (69) by solving (70)

So, now look at homogeneous IVP:

$$
\begin{aligned}
Y'' &= p(x)Y' + q(x)Y \\
Y(x_0) &= \alpha \\
Y'(x_0) &= s
\end{aligned}
\tag{71}
$$

Call its solution $Y = Y(s;x)$. If p and q are continuous on $[x_0, b] \Rightarrow$ there's a unique solution to (71).

But recall that every solution of (69) or (71) must be expressible as a linear combination of 2 independent solutions $y_1(x)$ and $y_2(x)$ which satisfy, say,

$$
\begin{array}{ll}
Y_1(x_0) = 1 & Y_1'(x_0) = 0 \\
Y_2(x_0) = 0 & Y_2'(x_0) = 1
\end{array}
\tag{72}
$$

(these ensure linear independence of Y_1 and $Y_2(x)$ and is not the only choice ... cf. Ch 11 of Boyce and DiPrima) so $Y(s;x) = \alpha Y_1(x) + sY_2(x)$ is the unique solution that satisfies $Y(x_0) = \alpha, Y'(x_0) = s$

Now take s such that

$$
Y(s;b) \equiv \alpha Y_1(b) + sY_2(b) = \beta \tag{73}
$$

then $Y(x) = Y(s;x)$ is a solution of BVP (69). Solving for

$$
s = \frac{\beta - \alpha Y_1(b)}{Y_2(b)} \quad ,
$$

ok provided $Y_2(b) \neq 0$.

Suppose $Y_2(b) = 0 \Rightarrow$ there may not be a solution to (69): if $Y_2(b) = 0$ a solution would exist **only if**

$$
\beta = \alpha Y_1(b) \quad \text{(see (73))}
$$

but it would not be unique since in this case

$$
Y(s;x) \text{ of (71) is a solution for } \underline{\text{arbitrary}}\ s.
$$

$\therefore$ There are 2 mutually exclusive cases: either a unique solution exists or else the "homogeneous problem," which is

$$(74) \qquad \begin{cases} Y'' = p(x)Y' + q(x)Y \\ Y(x) = 0 \\ Y(b) = 0, \end{cases}$$

has a nontrivial solution $sY_2(x)$, i.e. if $Y_2(x)$ is nontrivial solution then $Y_2(x)$ times any constant is also a nontrivial solution to (74) $\therefore$ there's a whole family of solutions (infinite family).

$\therefore$ either (69) has a unique solution OR ELSE (74) has a non-trivial solution. This is "The Alternative Principle." $\square$

The Shooting Method

Linear Case:

Take

$$(75) \qquad Y'' = p(x)Y' + q(x)r \quad x_0 \le x \le b \quad Y(x_0) = \alpha,\ Y'(x_0) = 0$$

and

$$(76) \qquad Y'' = p(x)Y' + q(x)Y \quad x_0 \le x \le b \quad Y(x_0) = 0,\ Y'(x_0) = 1.$$

If p, q, r continuous and $q > 0$ on $[x_0, b]$ then the Lipschitz condition exists for cast as a system $\Rightarrow$ both (75) and (76) have unique solutions.

Take $Y_1(x)$ solution of (75) and $Y_2(x)$ solution of (76)

$$(77) \qquad \Rightarrow Y(x) = Y_1(x) + \frac{\beta - Y_1(b)}{Y_2(b)} Y_2(x),$$

(provided $Y_2(b) \neq 0$. can be checked to be unique solution of

$$(78) \qquad BVP \equiv \begin{cases} Y'' = p(x)Y' + q(x)Y + r(x) \quad x_0 \le x \le b \\ Y(x_0) = \alpha \\ Y(b) = \beta \end{cases}$$

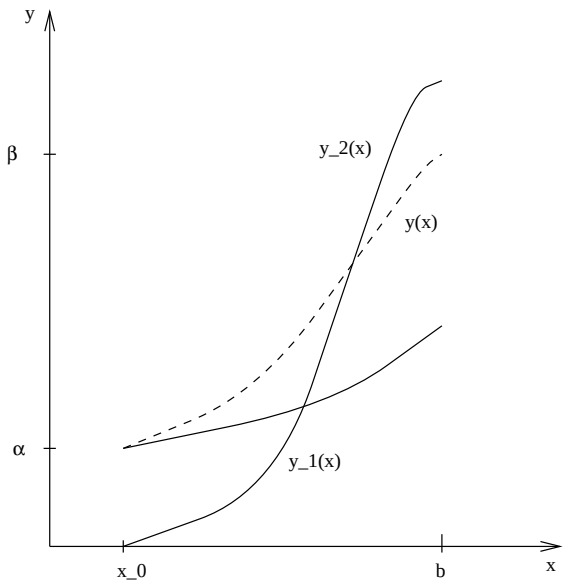

Figure 13: Graphical construction of the solution.

Remark: Note that if Y_2 is solution of $Y'' = p(x)Y' + q(x)Y$ and $Y_2(x_0) = Y_2(b) = 0 \Rightarrow Y_2 = 0$.

Summary:

So the shooting-method strategy amounts to the following: Replace (78) by 2 IVP (75) and (76). Use appropriate method to solve (75) and (76) and piece solution as per (77). Figure (13) shows graphically the construction of the solution $Y(x)$ in terms of $Y_1(x)$ and $Y_2(x)$.

ALGORIHM (from Burden and Faires p 582)

1. Set $h = (b - x_0)/N$
   $u_{1,0} = \alpha$
   $u_{2,0} = \alpha$
   $v_{1,0} = 0$
   $v_{2,0} = 1$
2. for $i = 0 \ldots N - 1$

$$\left[\begin{array}{l}\text{Use ERK4 (or some other suitable IVP scheme)}\\ \text{to solve for}\\ u_{1,i+1}, u_{2,i+1}, v_{1,i+1}, v_{2,i+1}\end{array}\right.$$

$$\begin{aligned} w_{1,0} &= \alpha \\ w_{2,0} &= \frac{\beta - u_{1,N}}{v_{1,N}} \end{aligned}$$

output $(x_0; w_{1,0} w_{2,0})$
here $w_{1,0}$ is an approximation to $Y(x_0)$ and $w_{2,0}$ an approximation to $Y'(x_0)$

$$\begin{aligned} \text{for } i &= 1, \cdots N \\ x &= x_0 + ih \\ w_1 &= u_{1,i} + w_{2,0} v_{1,i} \\ w_2 &= u_{2,i} + w_{2,0} v_{2,i} \\ \text{output } (x, w1, w2) & \\ & w_2 \text{ is an approximation to } Y'(x_i) \\ & \text{ and } w_1 \text{ is an approximation to} Y(x_i) \end{aligned}$$

END

□

The Shooting Method, Nonlinear Case

Similar to linear case, but cannot piece solution as linear combination of 2 IVP. Instead, we create a sequence of IVP's of the form

$$\left\{\begin{array}{l} y'' = f(x, y, y') \\ x_0 \le x \le b \\ y(x_0) = \alpha \\ y'(x_0) = t \end{array}\right. \tag{79}$$

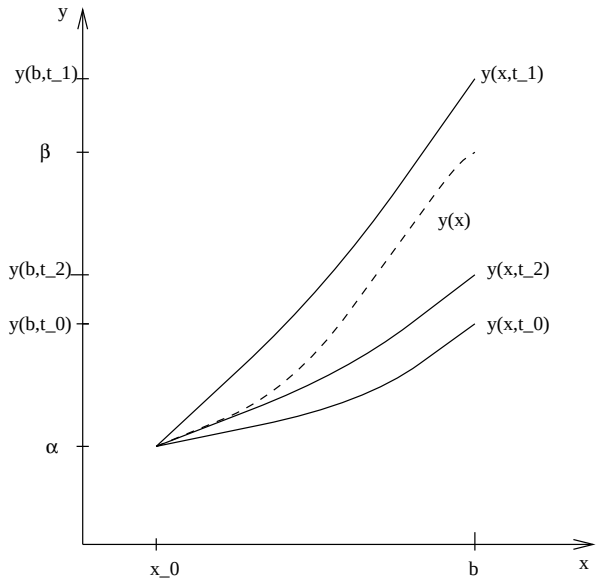

Figure 14: Nonlinear shooting method

t is a PARAMETER, chosen so that $t = t_k$, such that

$$\lim_{k\to\infty} y(b, t_k) = y(b) = \beta$$

$$1^{st} \text{ shot: result of} \begin{cases} y'' = f(x, y, y') & x_0 \le x \le b \\ y(x_0) = \alpha \\ y(x_0) = t_0 \end{cases}$$

If $y(b, t_0)$ not close enough to β, we choose another "elevation" t_1, and check to see if close enough. If not, choose the next "elevation" $t_2, \cdots$ until our "shots" get close to β. The situation is depicted in Figure (14), which clearly shows why the method bears its name.

How to choose t_k? If $y(x, t)$ is approx solution to (79) (The IVP) then we need to determine t such that

$$y(b, t) - \beta = 0$$

A nonlinear equation that can be solved using an efficient root-finding method

For example, using secant:

$$t_k = t_{k-1} - \frac{(y(b, t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{y(b, t_{k-1}) - y(b, t_{k-2})} \qquad k = 2, 3, \ldots$$

A better and more elegant method uses the Newton method integrated into the IVP sequence. See Burden and Faires for algorithm page 587.

The shooting method, when it works, is usually quite fast. It is easy to implement. Its good qualities are offset by possible instabilities. An alternative method, such as finite difference method, to be shown next.

Finite Difference Technique

LINEAR CASE

$$\text{Take} \begin{cases} Y'' = p(x)Y' + q(x)Y + r(x) & x_0 \le x \le b \\ Y(x_0) = \alpha \\ Y(b) = \beta \end{cases}$$

RECIPE

(Equally spaced grid case)

$$\begin{aligned} &\Rightarrow [x_0, b] && \text{and divide into } N+1 \text{ intervals} \\ &x_i = x_0 + ih && i = 0, 1, \cdots N+1 \\ &h = \frac{(b - x_0)}{N} \end{aligned}$$

Approximate Y'' and Y' by difference quotients

Approximate Y_n by y_n

Generate an $N \times N$ matrix problem for the unknowns y_n

The boundary data is at x_0, at which point $y_0 = \alpha$, and at x_{N+1}, at which point $y_{N+1} = \beta$.

Example In this instance we'll use low order center-differenced approximations to the derivatives. Assume that $Y \in C^4[x_{i+0}x_{i+1}]$. To get the finite

difference expressions to the derivatives, expand

$$Y(x_i \pm h) = Y(x_{i\pm 1}) = Y(x_i) \pm hY'(x_i) + \frac{h^2}{2}Y''(x_i) \pm \frac{h^3}{6}Y'''(x_i) + \frac{h^4}{24}Y''''(\xi_i) \tag{80}$$

for some $\xi_i^{\pm}$ in $(x_i, x_{i\pm 1})$

Add $Y(x_i + h)$ and $Y(x_i - h)$ expressions and get

$$Y''(x_i) = \frac{1}{h^2}\left[Y(x_{i+1}) - 2Y(x_i) + Y(x_{i-1})\right] - \frac{h^2}{24}\left[Y''''(\xi_i^+) + Y''''(\xi_i^-\right]$$

Use the Intermediate Value Theorem and (80) to deduce

$$Y''(x_i) = \frac{1}{h^2}\left[Y(x_{i+1}) - 2Y(x_i) + Y(x_{i-1})\right] - \frac{h^2}{24}Y''''(\xi_i)$$

with

$$\xi_i \in (x_{i-1}, x_{i+1})$$

So $Y''(x_i) \approx \frac{1}{h^2}\left[Y_{i+1} - 2Y_i + Y_{i-1}\right]$. This is a "centered-difference" aproximation to $Y''(x_i)$ and is an $\mathcal{O}(h^2)$

Exercise Show that

$Y'(x_i) \approx \frac{1}{2h}\left[Y_{i+1} - Y_{i-1}\right]$ is $O(h^2)$ approximation to Y' at $x = x_i$. The truncation error is $\frac{h^2}{6}Y''(\eta_i)$

□

So, now we project the equation onto our grid, with $y_i \equiv y(x_i)$, we get

$$\begin{cases} \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = p(x_i)\left[\dfrac{y_{i+1} - y_{i-1}}{2h}\right] + q(x_i)y_i + r(x_i) \quad i = 1, 2 \cdots N \\ y_0 = \alpha \quad y_{N+1} = \beta \end{cases} \tag{81}$$

The truncation error is $-\frac{h^2}{12}\left[2p(x_i)y'''(\eta_i) - y''''(\xi_i)\right] = \mathcal{O}(h^2)$. So (81) has the form of the linear system

$$A\mathbf{y} = \mathbf{b} \tag{82}$$

with

$$A = \begin{bmatrix} 2+h^2q_1 & -1+\frac{h}{2}p_1 & 0 & 0 & 0 \\ -1-\frac{h}{2}p_2 & 2+h^2q_2 & -1+\frac{h}{2}p_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & -1-\frac{h}{2}p_{N-1} & 2+h^2q_{N-1} & -1+\frac{h}{2}p_{N-1} \\ 0 & 0 & 0 & -1-\frac{h}{2}p_N & 2+h^2q_N \end{bmatrix}$$

and

$$\mathbf{y} = \begin{bmatrix} y_i \\ : \\ y_N \end{bmatrix} \mathbf{b} = \begin{bmatrix} -h^2r_1 + \left(1+\frac{h}{2}p_1\right)\alpha \\ -h^2r_2 \\ : \\ -h^2r_{N-1} \\ -h^2r_N + \left(1-\frac{h}{2}pN\right)\beta \end{bmatrix}$$

Theorem

Suppose p, q, r are continuous on $[x_o, b]$. If $q(x) \geq 0$ on $[x_0, b] \Rightarrow$ (82) has a unique solution provided $h < 2/L$ where $L = \max\limits_{x_0 \leq x \leq b} |p(x)|$. **Proof** (exercise). Look at conditions for the solution of the associated problem (82). □ **Remark:** How could we get higher than $\mathcal{O}(h^2)$ truncation error? Could use higher order approximation to derivatives, but this leads to more computing (nothing to be scared about these days). However, the more important problem is that it leads to a mathematical issue to resolve: Suppose you use $\mathcal{O}(h^4)$ approximation to the derivatives: we'll require knowledge of field at $y_{i-2}, y_{i-1}, y_i, y_{i+1}y_{i+2}$. In the interior this is not a problem...our matrix A will simply have a larger bandwidth. However, at the end-points we have some figuring out to do: at $i = 1$, we would need y_{-1}, y_0, y_1, y_2. The node y_0 is given by boundary conditions, y_1 and y_2 are interior points, so these are ok, but we don't have an equation or condition to determine y_1. At $i = N$, we have the same sort of problem but there the undertermined node is at y_{N+2}. As you might guess, using even a higher order scheme will generate even more undetermined boundary points. So, how do we deal with this problem? If we want to get high order accuracy we need to determine these unknowns.

Strategies:

- Use a right-sided finite difference approximation to derivatives at $i = 0$ and a left-sided finite difference approximation to derivatives at $i = N + 1$. This seldom works. What you need to do is construct the associated linear algebra problem, (82), and check to see if the system indeed has a solution...what happens in many instances is that you get an underdetermined system.

- Use a lower order discretization and commensurably smaller step sizes at the ends and match the solution there to the interior so that the overall order of the scheme is retained. This is a rather clumsy strategy and sometimes generates very fragile code. It will only work in certain instances. In fact, in some instances, for example, in boundary layer problems, a sudden change in grid size will generate spurious and usually noisy approximations at the end points.

- Use the physics of problem to generate conditions for all undetermined y_i's to the left and the right of $i = 1$ and at $i = N$, repectively. This is a very sensible approach, but is not always possible.

- Use Richardson extrapolation (e.g. stick to the second-order approximation and you get higher order but you'll have to compute the problem twice). This is also a very sensible approach and is viable, provided you have precise estimates for the truncation error (not a difficult thing to calculate). Remember, though, that as a rule you don't want to use extrapolation more than a couple of times, since it may become too computationally intensive, as compared to simply using a higher order scheme, or you may encounter rounding-errors. See Richardson Extrapolation.

- Combine the last two strategies.

Unequal Spacing: One other possibility for dealing with the above problem is to use variable-sized meshes. This is not only used to get around the problem of undetermined quantities but is also generally applicable to boundary value problems in which multiple scales in the approximate solution are expected to arise. This is typically used to solve problems which have localized values of x_i for which a lot of changes in the dependent variables are seen, but not much is happening in other areas. This is a topic of current research and is beyond the scope of presentation: in general, one hope to obtain higher

resolution in certain places where y varies a lot and go with low resolution in places where y does not change all that much. The overall aim is to reduce the computational expense of the method, as compared to over-resolving the whole domain, but this must be done with care. Perhaps the most research in this area is done by the finite element methods community and if you are interested, you could start by just finding out what finite element methods are. Then look into adaptive mesh refinement schemes and domain decomposition methods.

NONLINEAR CASE:

Take

$$\text{(83)} \qquad \begin{cases} Y'' = f(x, Y, Y') & x_0 \le x \le b \\ Y(x_0) = \alpha \\ Y(b) = \beta \end{cases}$$

In principle, it's simple. Same deal as above, but now we get a nonlinear system of algebraic equations. How to solve? Use Newton's method or fixed point iteration. But before launching into either of these two solution techniques, make sure that you study the system you're solving: does the (83) have a solution? Is it unique? Will a fixed point method be applicable? Again, fixed point methods will not require the calculation of an analytical or approximate Jacobian.

The easy ones are those with a unique solution, i.e. when

$$\text{(84)} \qquad \begin{cases} \text{(i)} & f, f_Y, f'_Y \text{ are all continuous on} \\ & D = \{(x, Y, Y') | (x_0 \le x \le b), -\infty < Y, Y' < \infty\} \\ \text{(ii)} & f_Y(x, y, y') \ge \delta > 0 \text{ on } D \text{ for some } \delta > 0 \\ \text{(iii)} & k \text{ and } L \text{ exist (constants), such that} \\ & k = \max_D |f_Y(x, Y, Y')| \quad L = \max_D |f_{Y'}(x, Y, Y')| \end{cases}$$

□

Recipe: Using $\mathcal{O}(h^2)$ discretization for Y, its derivatives, and the equation coefficients, just as in linear case.

1. Form associated nonlinear algebraic system.

2. Solve system using fixed point methods, or better yet, express the resulting nonlinear system of algebraic equations as

$$\mathbf{S}(\mathbf{y}^{\ell+1}) = 0$$

with $\mathbf{y}^{\ell}= \begin{bmatrix} y_1^{\ell} \\ \vdots \\ . \\ y_N^{\ell} \end{bmatrix}$, known. The superscript ℓ is the iteration counter.

then $\mathbf{S}(\mathbf{y}^{\ell+1}) = \mathbf{S}(\mathbf{y}^{\ell}) + J(\mathbf{y}^{\ell})\delta\mathbf{y}^{\ell} + \mathcal{O}(\delta\mathbf{y^2}) = 0$ assume $\mathcal{O}(\delta y^2)$ small, then the approximation

$$J(y^{\ell})\delta y^{\ell} = -\mathbf{S}(y^{\ell}) \tag{85}$$

where $J(\mathbf{Y}^{\ell})$ is the Jacobian of the nonlinear algebraic problem, evaluated at $\mathbf{y}^{\ell}$, may be suitable. And $\delta\mathbf{y}^{\ell}$ is the correction on $\mathbf{y}^{\ell}$, so that

$$\mathbf{y}^{\ell+1} = \mathbf{y}^{\ell} + \delta\mathbf{y}^{\ell}$$

Problem (85) is a linear algebraic problem, in fact, it can be shown to be tridiagonal.

Implementation comments: don't forget to put a "max iterations exceeded" condition in your code. Also, since scheme is $\mathcal{O}(h^2)$, use $\mathcal{O}(h^2)$ stopping criteria for the Newton iteration. You can use fixed point method in the iteration, but convergence is linear and has restrictions on the type of problem for which it is applicable (see Fixed Point iteration. Since a good intial guess is required, make it go through (x_0, α) and (b, β) and in addition, satisfy appropriate conditions as per (83) on previous page.

Want to use a higher order truncation scheme? Don't neglect considering a low-order method, coupled with Richardson extrapolation, before getting into something more involved. Otherwise, use a higher-order scheme, but make sure that you don't leave any boundary points undertermined.

□

0.2.1 The Elliptic Problem with Discontinuities

We consider here

$$-\frac{d}{dx}(a(x)\frac{du(x)}{dx}) + c(x)u(x) = f(x) \qquad a \le x \le b, \tag{86}$$

with boundary conditions

$$\begin{aligned} \alpha_1 u(0)\beta_1 u'(0) &= \gamma_1, \qquad \alpha_1, \beta_1 \ge 0, \quad \alpha_1 + \beta_1 > 0 \\ \alpha_2 u(1)\beta_2 u'(1) &= \gamma_2, \qquad \alpha_2, \beta_2 \ge 0, \quad \alpha_2 + \beta_2 > 0. \end{aligned} \tag{87}$$

This is the general elliptic problem in 1 dimension. We already have covered how one could solve this equation using simple finite difference equations, if $a(x) > 0$ is smooth and continuous. Try setting up the linear algebraic problem using naive finite differences when $a(x)$ is piece-wise continuous and has jumps, taking $c(x) = 0$ for simplicity. What happens?

Here we present what is sometimes called the "box method". Take a, c, f piecewise continuous in $a \le x \le b$, with $a(x) > 0$, $c(x) \ge 0$ for all $a \le x \le b$.

If $a(x)$ is discontinuous at some discrete places x_j, then we exploit the fact that $u(x)$ and the flux $a(x)\frac{du(x)}{dx}$ is continuous at these points and everywhere else. Take one of these discontinuity locations, we enforce

$$a(x_j^+)\frac{du_j^+}{dx} = a(x_j^-)\frac{du_j^-(x)}{dx}, \tag{88}$$

where $\pm$ refers to the right/left side of the discontinuity. First, let's create a mesh $a = x_0 < x_1, ... < x_{I+1} = b$, and the mesh spacing is $h_i = x_{i+1} - x_i$. Integrating (86) over $[x_i, x_i + h_i/2] = [x_i, x_{i+1/2}]$ yields

$$-a_{i+1/2}\frac{du_{i+1/2}}{dx} + a(x_i^+)\frac{du(x_i^+)}{dx} + \int_{x_i}^{x_{i+1/2}} c(x)u(x)dx = \int_{x_i}^{x_{i+1/2}} f(x)dx. \tag{89}$$

Integrating over $[x_{i-1/2}, x_i]$ yields

$$a_{i-1/2}\frac{du_{i-1/2}}{dx} - a(x_i^-)\frac{du(x_i^-)}{dx} + \int_{x_i-1/2}^{x_i} c(x)u(x)dx = \int_{x_{i-1/2}}^{x_i} f(x)dx. \tag{90}$$

Adding (89) and (90), employing (88) gives

$$-a_{i+1/2}\frac{du_{i+1/2}}{dx}+a_{i-1/2}\frac{du_{i-1/2}}{dx}+\int_{x_{i-1/2}}^{x_{i+1/2}}c(x)dx=\int_{x_{i-1/2}}^{x_{i+1/2}}f(x)dx \tag{91}$$

which is exact for the interval $[x_{i-1/2}, x_{i+1/2}]$.

0.2.2 Discrete approximation for interior mesh points

Center-differencing derivatives and midpoint rule over half intervals of (91) leads to

$$-a_{i+1/2}\frac{u_{i+1}-u_i}{h}+a_{i-1/2}\frac{u_i-u_{i-1}}{h}+u_i\frac{1}{2}(c_i^-h_{i-1}+c_i^+h_i)=\frac{1}{2}(f_i^-h_{i-1}+f_i^+h_i)+=\frac{1}{2}(h_{i-1}+h_i)\tau_i. \tag{92}$$

If a, c, f, u are smooth $\tau_i = (\overline{h}_i)$, where $\overline{h}_i = \max(h_i, h_{i-1})$, if $h_i \neq h_{i-1}$. $\tau_i = (\overline{h}_i^2)$, where $\overline{h}_i = \max(h_i, h_{i-1})$, if $h_i = h_{i-1}$, which follows from a Taylor series expansion of u_{i-1}, u_{i+1} and $c_i = c_i^- = c_i^+, f_i = f_i^- = f_i^+$, expanded about u_i.

0.2.3 Discrete Approximation at Boundary Points

Consider (89):

$$-a_{i+1/2}\frac{du_{i+1/2}}{dx}+a(x_i^+)\frac{du(x_i^+)}{dx}+\int_{x_i}^{x_{i+1/2}}c(x)u(x)dx=\int_{x_i}^{x_{i+1/2}}f(x)dx, \tag{93}$$

in the interval $[x_0, x_2]$: At $x_0 = a$ we have:

$$\begin{aligned}\alpha_1u(a)-\beta_1u'(a) &= \gamma_1, \qquad \text{so,}\\ \frac{\alpha_1u(a)-\gamma_1}{\beta_1} &= u'(a) \qquad \beta_1\neq 0\\ u(a) &= \frac{\gamma_1}{\alpha_1} \qquad \beta_1=0.\end{aligned} \tag{94}$$

At the other end point

$$\begin{aligned} \alpha_2 u(b) - \beta_2 u'(b) &= \gamma_2, \qquad \text{so,} \\ \frac{\alpha_2 u(b) - \gamma_2}{\beta_2} &= u'(b) \qquad \beta_2 \neq 0 \\ u(b) &= \frac{\gamma_2}{\alpha_2} \qquad \beta_2 = 0. \end{aligned} \tag{95}$$

Dirichlet Case $\beta_1 = 0$: In this case, $u_0 = u(0)$ is known, and the discrete equations at u_0 and u_1 are:

$$u_0 = \gamma_1/\alpha_1,$$

which is exact, or (92) at $i = 1$, $\mathcal{O}(\bar{h})$ or$\mathcal{O}(\bar{h}^2)$.

Neumann (or mixed) Case $\beta_1 \neq 0$: In this case $u'(a)$ is known, but $u(0) = u_0$ is unknown, and (89) is employed for $u(0)$:

$$-a_{1/2}\frac{u_1 - u_0}{h_0} + a_0^+ \frac{\alpha_1 u_0 - \gamma_1}{\beta_1} + c_0^+ \frac{u_0 h_0}{2} = \frac{f_0^+ h_0}{2} + \frac{1}{2} h_0 \tau_0, \tag{96}$$

or (92) at $i = 1$, $\mathcal{O}(\bar{h})$ or$\mathcal{O}(\bar{h}^2)$.

The discrete equations are straightforward to write down in linear-algebraic form, the resulting matrix is *symmetric*, tridiagonal. This is the case for both the Dirichlet and Neumann cases. The equations at the first, last and interior grid points are:

$$\begin{aligned} u_0 &= \frac{\gamma_1}{\alpha_1} \\ u_{I+1} &= \frac{\gamma_2}{\alpha_2} \\ -a_{i+1/2}\frac{u_{i+1} - u_i}{h_i} + a_{i-1/2}\frac{u_i - u_{i-1}}{h_{i-1}} + u_i \frac{c_i^- h_{i-1} + c_i^+ h_i}{2} &= \frac{f_i^- h_{i-1} + f_i^+ h_i}{2}. \end{aligned}$$

For the Neumann or Mixed case, the first, last, and interior points are:

$$\begin{aligned} -a_{1/2}\frac{u_1 - u_0}{h_0} + a_0^+ \frac{\alpha_1 u_0 - \gamma_1}{\beta_1} + c_0^+ \frac{u_0 h_0}{2} &= \frac{f_0^+ h_0}{2}. \\ -a_{I+1}^- \frac{\gamma_2 - \alpha_2 u_{I+1}}{\beta_2} + a_{I+1/2}\frac{u_{I+1} - u_I}{h_I} + c_{I+1}^- \frac{u_{I+1} h_I}{2} &= \frac{f_{I+1}^+ h_I}{2}. \\ -a_{i+1/2}\frac{u_{i+1} - u_i}{h_i} + a_{i-1/2}\frac{u_i - u_{i-1}}{h_{i-1}} + u_i \frac{c_i^- h_{i-1} + c_i^+ h_i}{2} &= \frac{f_i^- h_{i-1} + f_i^+ h_i}{2}. \end{aligned}$$

□

0.2.4 The Method of Weighted Residuals (MWR)

Consider the well–posed differential equation

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^n$$

with boundary conditions on $\partial\Omega$. We will approximate the solution $u(\mathbf{x})$ as

$$v(\mathbf{x}) = \sum_{j=1}^{N} c_j \phi_j(\mathbf{x})$$

where $\{\phi_j\}_{j=1}^N$ span the "trial space". Hence, we denote $v(\mathbf{x})$ as the "trial functions". The goal in MWR is the determination of the N scalars $\{c_j\}_{j=1}^N$.

This is done in MWR by minimizing

$$r(\mathbf{x}) \equiv \mathcal{L}v(\mathbf{x_0} - f(\mathbf{x})$$

where $r(\mathbf{x})$ is the "residual". We do so by attemping to find the coefficients that drive the weighted average

$$\int_\Omega r(\mathbf{x}) w_i(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0 \qquad \forall i,$$

where $\{w_i\}_{i=1}^M$ are the "test functions" or weights. The number of weight functions and N are related as shown below.

A good choice of basis functions are the Lagrange poynomials. To be specific, let $n = 1$ and $\{x_j\}_{j=1}^N$ be the set of N "nodes" on some intreval $[x_0, x_f] \equiv \Omega$, they are distinct but not necessarily evenly spaced. Here x_0 and x_f are the boundary points $\partial\Omega$.

An $(N-1)^{th}$ degree polynomial associalted with x_j, $j = 1, \ldots, N$

$$\ell_j(x) = \prod_{i=1, i\neq j}^{N} \frac{x - x_i}{x_j - x_i}$$

forms a basis of S_N, the finite-dimensional linear space. As we saw in 475A

$$v_i \equiv v_(x_i) = \sum_{j=1}^{N} c_j \ell_j(x_i) = c_i \ell_i(x) = c_i,$$

since $\ell_j(x_i) = \delta_{ij}$. So we have

$$v(x) = \sum_{j=1}^{N} v_j \phi_j(x)$$

where the ϕ_j are Lagrange polynomials.

These $\ell_j(x)$ are nonzero over entire Ω, except at a finite number of node points. Further, $\ell_j(x) \in C^\infty(\Omega)$. However, we may want to choose piecewise polynomials rather than global polynimials. This is beneficial for parallel computing where we want to minimze communication between processors.

The following sets of piecewise defined polynomials in $C^0(\Omega)$ are simple and thus quite popular.

- piecewise linear

$$\phi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}} & x_{j-1} \le x \le x_j \\ \frac{x_{j+1}-x_j}{x_{j+1}-x_j} & x_j \le x \le x_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

  Download the piecewise linear polynomial interpolation matlab script.

- piecewise quadratic and still $C^0(\Omega)$

  j odd

$$\phi_j(x) = \begin{cases} \frac{(x-x_{j-1})(x-xj-2)}{(x_j-x_{j-1})(x_j-x_{j-2})} & x_{j-2} \le x \le x_j \\ \frac{(x_{j+1}-x)(x_{j+2}-x)}{(x_{j+1}-x_j)(x_{j+2}-x_j)} & x_j \le x \le x_{j+2} \\ 0 & \text{otherwise} \end{cases}$$

  j even

$$\phi_j(x) = \begin{cases} \frac{(x-x_{j-1})(x_{j+1}-x)}{(x_j-x_{j-1})(x_j+1-x_j)} & x_{j-1} \le x \le x_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

Download the piecewise quadratic interpolation matlab script. Compare to the piecewise linear and the cubic interpolation cases (see below).

- for $C^1(\Omega)$ functions use piecewise Hermite polynimoals

$$v(x) = \sum_{j=1}^{N} v_j \phi_{oj}(x) + \frac{\mathrm{d}v_j}{\mathrm{d}x} \phi_{1j}(x) \,.$$

Here v_j is the approximation u at x_j, $\frac{\mathrm{d}v_j}{\mathrm{d}x}$ is the approximation of $\frac{\mathrm{d}u}{\mathrm{d}x}$ at x_j and ϕ_{0j} and ϕ_{1j} are piecewise Hermite polynomials. Here are the cubic Hermite polynomials:

$$\phi_{0j}(x) = \begin{cases} \frac{(x-x_{j-1})^2[2(x_j-x)+(x_j-x_{j-1}]}{(x_j-x_{j-1})^3} & x_{j-1} \le x \le x_j \\ \frac{(x-x_{j+1})^2[2(x-x_j)+(x_{j+1}-x_j)]}{(x_{j+1}-x_j)^3} & x_j \le x \le x_{j+1} \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

$$\phi_{1j}(x) = \begin{cases} \frac{(x-x_{j-1})^2(x-x_j)}{(x_j-x_{j-1})^2} & x_{j-1} \le x \le x_j \\ \frac{(x-x_{j+1})^2(x-x_j)}{(x_{j+1}-x_j)^2} & x_j \le x \le x_{j+1} \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

with the properties

$$\begin{aligned} \phi_{0j}(x_i) &= \delta_{ij} \qquad & \phi_{0j}(x_i) &= 0 \\ \frac{\mathrm{d}\phi_{0j}}{\mathrm{d}x}(x_i) &= 0 \qquad & \frac{\mathrm{d}\phi_{1j}}{\mathrm{d}x}(x_i) &= \delta_{ij} \,. \end{aligned}$$

Download the piecewise cubic Hermite polynomial interpolation matlab script. Verify the properties of the interpolants and compare the results of the cubic interpolation to the quadratic and linear cases.

Recall that MWR goal is to minimize $r(x)$ by forcing it to zero in a weighted average sense over the domain Ω. The most popular variants of MWR are

1. Subdomain Method

2. Collocation

3. Galerkin and Petrov–Galerkin

We will only present the first 2 by example.

0.2.5 Subdomain Method

We minimize the residual $r(x)$ by forcing it to zero in a weighted average sense as follows: minimize $r(x)$ over Ω by forcing the arithmetic average of $r(x)$ taken over discrete intervals of Ω to be zero.

Choose weights

$$w_i(x) = \begin{cases} 1 & x \in \Omega_i \\ 0 & x \neq \Omega_i \end{cases}$$

where $\{\Omega_i\}_{i=1}^M$ are monimtersecting subregions within Ω whose union covers Ω. For piecewise linear on nodes $\{x_i\}_{i=1}^N$ a good choice of Ω_i is

$$(x_i - \frac{1}{2}(x_i - x_{i-1})) < x < (x_i + \frac{1}{2}(x_{i+1} - x_i))$$

. In this case

$$w_i(x) = \begin{cases} 1 & x_i - \frac{\nabla x_i}{2} < x < x_i + \frac{\Delta x_i}{2} \\ 0 & \text{otherwise} \end{cases}$$

where $\nabla x_i = x_i - x_{i-1}$ and $\Delta x_i = x_{i+1} - x_i$. When x_i is near a boundary node the Ω_i is taken as only the region residing within Ω.

Example:

$$\mathcal{L}u = \left(\frac{\mathrm{d}^2}{\mathrm{d}x^2} + k^2\right) u = 0 \qquad 0 < x < 1$$
$$u(0) = 1 \qquad u(1) = 0$$

k is a given real number. The exact solution to the problem is
$u(x) = \cos(kx) - \cot(k)\sin(kx)$.

Let us use piecewise linear functions and the subdomain method. Choose $x_1 = 0$, $x_2 = 0.5$ and $x_3 = 1.0$, here $\Delta x = 0.5$.

We want to have

$$\int_0^1 r(x) w_1 \, \mathrm{d}x = 0,$$
$$\int_0^1 r(x) w_2 \, \mathrm{d}x = 0,$$
$$\int_0^1 r(x) w_3 \, \mathrm{d}x = 0.$$

Since $v_1 = v(0) = 1$ and $v_3 = v(1) = 0$ and $f = 0$. So we have only to ensure that

$$\int_0^1 r(x) w_2 \, \mathrm{d}x = \int_0^1 \left(\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} + k^2 v \right) w_2(x) \, \mathrm{d}x = 0$$

or

$$\sum_{j=1}^{3} v_j \int_{0.25}^{0.75} \left(\frac{\mathrm{d}^2 \phi_j}{\mathrm{d}x^2} + k^2 \phi_j \right) \mathrm{d}x = 0$$

Let's write

$$\frac{\mathrm{d}\phi_2}{\mathrm{d}x} = \frac{1}{\Delta x} H(x - 0) - \frac{2}{\Delta x} H(x - 0.5) + \frac{1}{\Delta x} H(x - 1)$$

where $H(x - x_p)$ is the Heaviside function

$$H(x - x_p) = \begin{cases} 1 & x \geq x_p \\ 0 & x < x_p \end{cases}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}x} H(x - x_p) = \delta(x - x_p)$$

where $\delta(x - x_p)$ is the Dirac delta function. This function is zero for $x \neq x_p$ and unbounded at $x = x_p$ such that we have

$$\int_{-\infty}^{\infty} \delta(x - x_p) \, \mathrm{d}x = 1$$
$$\int_{-\infty}^{\infty} \delta(x - x_p) F(x) \, \mathrm{d}x = F(x_p)$$

where $F(x)$ is a well–defined function. It follows that

$$\frac{\mathrm{d}^2\phi_j}{\mathrm{d}x^2} = \frac{1}{\Delta x}\delta(x-0) - \frac{2}{\Delta x}\delta(x-0.5) + \frac{1}{\Delta x}\delta(x-1)$$

and

$$v_1 \int_{0.25}^{0.75} \left[\frac{1}{\Delta x}\delta(x-0.5) + k^2\phi_1(x)\right] \mathrm{d}x + v_2 \int_{0.25}^{0.75} \left[-\frac{2}{\Delta x}\delta(x-0.5) + k^2\phi_2(x)\right] \mathrm{d}x + v_3 \int_{0.25}^{0.75} \left[\frac{1}{\Delta x}\delta(x-0.5) + k^2\phi_3(x)\right] \mathrm{d}x = 0$$

Since $\delta x = 0.5$

$$v_1\left[2 + \frac{k^2}{16}\right] + v_2\left[-4 + \frac{3k^2}{8}\right] + v_3\left[2 + \frac{k^2}{16}\right] = 0$$

Since $v_1 = 1$ and $v_3 = 0$ we find the following system of equations

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{32+k^2}{16} & \frac{-32+3k^2}{8} & \frac{32+k^2}{16} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Solving gives $(v_1, v_2, v_3) = (1, \frac{32+k^2}{64+6k^2}, 0)$.

Compare this to $(u_1, u_2, u_3) = (1, \cos(0.5k) - \cot(k)\sin(0.5k), 0)$.

0.2.6 Collocation Method:

Minimize residual by forcing it to pass through zero at a finite number of discrete points within Ω. So here

$$w_i(x) = \delta(x - x_i^c), \qquad x_i^c \in \Omega$$

where x_i^c is the i–th collocation point. The choice of collocation points is an important consideration in the method. We aim to have

$$\int_\Omega r(x)w_i(x)\,\mathrm{d}x = \int_\Omega r(x)\delta(x - x_i^c)\,\mathrm{d}x = r(x_i^c) = 0$$

In other words, at each collocation point the trial functions are required to satisfy the differential equation exactly. The number of collocation points is related to the number of c_j.

In general for an M-th order equation: choose polynomials of degree $(2M-1)$ with $C^{M-1}(\Omega)$ continuity. m collocation points are chosen within each element located at the roots of the M–th degree Legendre polynomial over each element.

Example: Solve

$$\mathcal{L}u = \left(\frac{\mathrm{d}^2}{\mathrm{d}x^2} + k^2\right) u = 0 \qquad 0 < x < 1$$
$$u(0) = 1 \qquad u(1) = 0$$

Because $r(x)$ involves second order derivatives of trial functions, the trial space must be at least $C^1(\Omega)$ for $r(x)$ to remain bounded.

We choose piecewise cubic Hermite polynomials

$$v(x) = \sum_{j=1}^{N} v_j \phi 0 j(x) + \frac{\mathrm{d}v_j}{\mathrm{d}x}\phi_{1j}(x)$$

Take $N = 2$. Nodes are located at $x = 0$ and $x = 1$. For second order differential equations with piecewise cubic Hermitian trial space, two collocation points should be chosen per element. Coupled with two boundary conditions we get $2N$ algebraic equations for $2N$ unknown nodal values. Further, if we choose the Gauss–Legendre quadrature points we obtain $O(\Delta x^4)$ accuracy. So the collocation points will be $x_1^c = (3-\sqrt{3})/6$ and $x_2^c = (3+\sqrt{3})/6$.

$$r(x_1^c) = \sum_{j=1}^{2} v_j \left.\frac{\mathrm{d}^2\phi_{0j}}{\mathrm{d}x^2}\right|_{x=x_1^c} + \left.\frac{\mathrm{d}v_j}{\mathrm{d}x}\frac{\mathrm{d}^2\phi_{1j}}{\mathrm{d}x^2}\right|_{x=x_1^c}$$
$$+ k^2\left[\sum_{j=1}^{2} v_j \left.\phi_{0j}\right|_{x=x_1^c} + \left.\frac{\mathrm{d}v_j}{\mathrm{d}x}\phi_{1j}\right|_{x=x_1^c}\right] = 0$$
$$r(x_2^c) = \sum_{j=1}^{2} v_j \left.\frac{\mathrm{d}^2\phi_{0j}}{\mathrm{d}x^2}\right|_{x=x_2^c} + \left.\frac{\mathrm{d}v_j}{\mathrm{d}x}\frac{\mathrm{d}^2\phi_{1j}}{\mathrm{d}x^2}\right|_{x=x_2^c}$$
$$+ k^2\left[\sum_{j=1}^{2} v_j \left.\phi_{0j}\right|_{x=x_2^c} + \left.\frac{\mathrm{d}v_j}{\mathrm{d}x}\phi_{1j}\right|_{x=x_2^c}\right] = 0$$

So the system to solve is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -5.657+0.998k^2 & -3.868+0.027k^2 & 5.657+0.002k^2 & -1.830-0.0008k^2 \\ 5.657+0.002k^2 & 1.830+0.0008k^2 & -5.657+0.998k^2 & 3.868-0.027k^2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ \frac{\mathrm{d}v_1}{\mathrm{d}x} \\ v_2 \\ \frac{\mathrm{d}v_2}{\mathrm{d}x} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

0.2.7 Galerkin Method:

Choose $w_i(x) = \phi_i(x)$, the basis of the trial space. If we used a different basis we have Petrov–Galerkin. So

$$\int_\Omega r(x)\phi_i(x0\,\mathrm{d}x = 0$$

ie we require that $r(x)$ be orthogonal to $\phi_i(x)$.

Example:

$$\mathcal{L}u = \left(\frac{\mathrm{d}^2}{\mathrm{d}x^2} + k^2\right) u = 0 \qquad 0 < x < 1$$
$$u(0) = 1 \qquad u(1) = 0$$

Again the nodes are $x = 0$, $x = 0.5$ and $x = 1$.

$$v(x) = \sum_{j=1}^{3} v_j \phi_j(x)$$

$v_1 = 1$ and $v_3 = 0$ by boundary conditions. So we only have to work out things for node 2:

$$\int_0^1 \left(\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} + k^2 v\right) \phi_2(x)\,\mathrm{d}x = 0$$

or

$$\sum_{j=1}^{3} v_j \int_0^1 \left(\frac{\mathrm{d}^2 \phi_j}{\mathrm{d}x^2} + k^2 \phi_j\right) \phi_2(x)\,\mathrm{d}x = 0$$

However we can rewrite using integration by parts

$$\int_0^1 \left(\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} + k^2 v \right) \phi_2(x)\, \mathrm{d}x = \int_0^1 \left(-\frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\phi_2}{\mathrm{d}x} + k^2 v\phi_2 \right) \mathrm{d}x + \left. \frac{\mathrm{d}v}{\mathrm{d}x}\phi_2 \right|_0^1$$

but $\phi_2 = 0$ at $x = 0$ and $x = 1$ so the last term is zero.

$$\begin{aligned}\int_0^1 \left(-\frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\phi_2}{\mathrm{d}x} + k^2 v\phi_2 \right) \mathrm{d}x &= \sum_{j=1}^{3} v_j \int_0^1 \left(-\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\frac{\mathrm{d}\phi_2}{\mathrm{d}x} + k^2 \phi_j \phi_2 \right) \mathrm{d}x \\ &= v_1 \left[\frac{1}{\Delta x} + \frac{k^2 \Delta x}{6} \right] + v_2 \left[-\frac{2}{\Delta x} + \frac{2k^2 \Delta x}{3} \right] + v_3 \left[\frac{1}{\Delta x} + \frac{k^2 \Delta x}{6} \right] = 0\end{aligned}$$

□

Another Comparative Example

Solve the

$$BVP \begin{cases} Y'' = 6t & 0 \le t \le 1 \\ Y(0) = 0 \\ Y(1) = 1 \end{cases}$$

The exact solution is clearly $Y(t) = t^3$. First, let's solve this problem using:

A) Collocation Technique

seek a function $u(t)$ that satisfies BVP at a discrete set of mesh points in interval. We choose $u(t)$ as a simple polynomial, capable of satisfying the boundary conditions and with the regularity suggested by the BVP.

For illustration → only 1 point $t = 0.5$ and $y = 0$, at $t = 1$.

$$\begin{array}{ll} \text{Pick} & u(t) = x_0 + x_1 t + x_2 t^2 \\ \text{so} & u'(t) = x_1 + 2x_2 t \\ \text{and} & u''(t) = 2x^2 \end{array}$$

So for

$$\begin{cases} Y'' = f(t, Y, Y') & x_0 \le t = b \\ Y(a) = \alpha \\ Y(b) = \beta \end{cases}$$

we require that $u = Y$ at 3 points requires 3 equations:

$$x_0 + x_1 a + x_2 a^2 = \alpha \tag{97}$$

$$x_0 + x_1 b + x_0 b^2 = \beta \tag{98}$$

equations (97) and (98) lead to

$$x_0 = 0$$
$$x_1 + x_2 = 1$$

and for some $t \in (a, b) \quad u''(t) = f(t, u(t), u'(t))$.

For us $u''(0.5) = f(0.5, u(0.5), u'(0.5))$

Thus

$$2x_2 = 6(0.5) = 3 \tag{99}$$

So solving (97), (98), (99) get $x_0 = 0$, $x_1 = -5$, $x_2 = 1.5$ thus the approximate solution is $u = -0.5t + 1.5t^2$. A comparison to the exact solution appears in Figure (15)

$$\begin{aligned} u(0.5) &= -0.5(0.5) + 1.5(0.5)^2 = (0.5)^2(-1 + 1.5) = (0.5)^3 \\ \text{compare to } Y(0.5) &= (0.5)^3 \quad \text{residual is 0} \end{aligned}$$

B) FEM/Galerkin Method:

Same BVP and use same 3 mesh points, which now become "knots" in the piecewise polynomial approximation. Take "hat" basis or elements, which are shown in Figure (16).

So $Y(t) \approx u(t) = x_1\phi(t) + x_2\phi_2(t) + x_3\psi(t)$.

Applying the boundary conditions,

$$\begin{aligned} u(0) &= 0 = x_1\phi_1(0) + x_2\phi_2(0) + x_3\phi_3(0) \Rightarrow x_1 = 0 \\ u(1) &= 1 = x_1\phi_1(1) + x_2\phi_2(1) + x_3\phi_3(1) \Rightarrow x_3 = 1 \end{aligned}$$

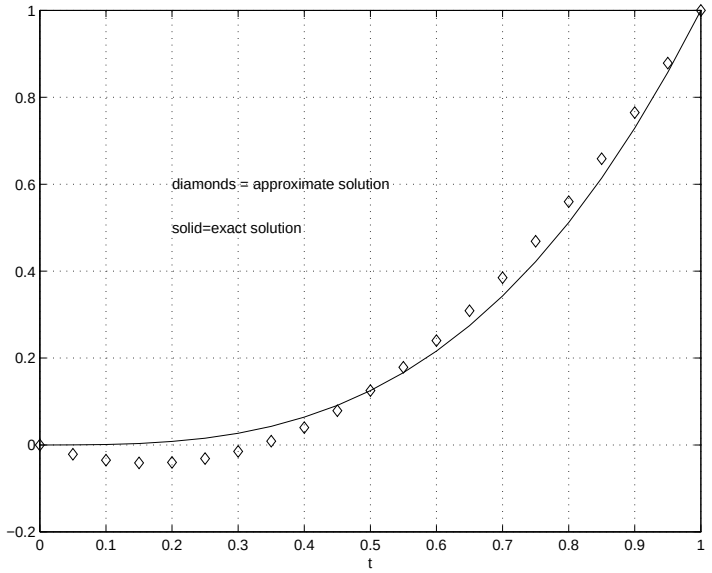

Figure 15: Comparison of exact and approximate solution via collocation

Galerkin condition applied at $t = 0.5 \Rightarrow$ residual must be orthogonal to space spanned by the basis functions and hence to each basis individually

$$\text{Orthogonality condition: } \int_0^1 (u''(t) - 6t)\phi_2(t)dt = \int_0^1 u''\phi_2(t)dt - 6\int_0^1 6\phi_2(t)dt \equiv 0$$

integrate by parts:

$$= -\int_0^1 u\overbrace{\phi_2'(t)dt + u'(t)\phi_2(t)}|_0^1 - \frac{3}{2} = 0$$

since $\phi_2(0) = \phi_2(1) = 0 \therefore 2^{nd}$ term drops out, thus,

$$= +\int_0^1 u'\phi_2'(t)dt + \frac{3}{2} = 0.$$

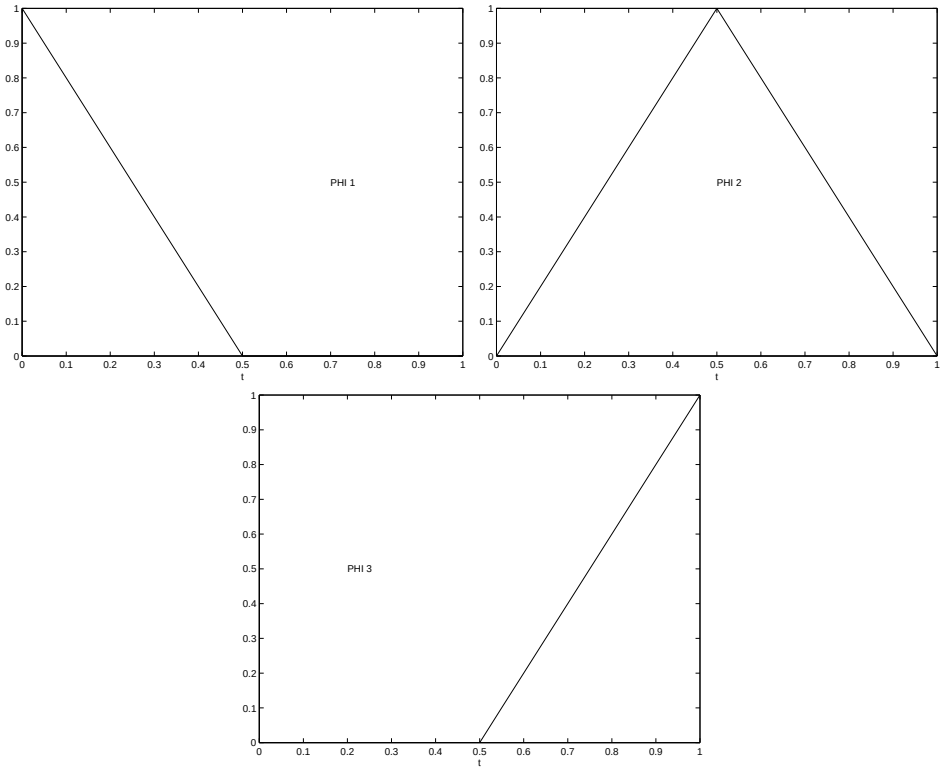

Figure 16: Hat functions, on the unit interval.

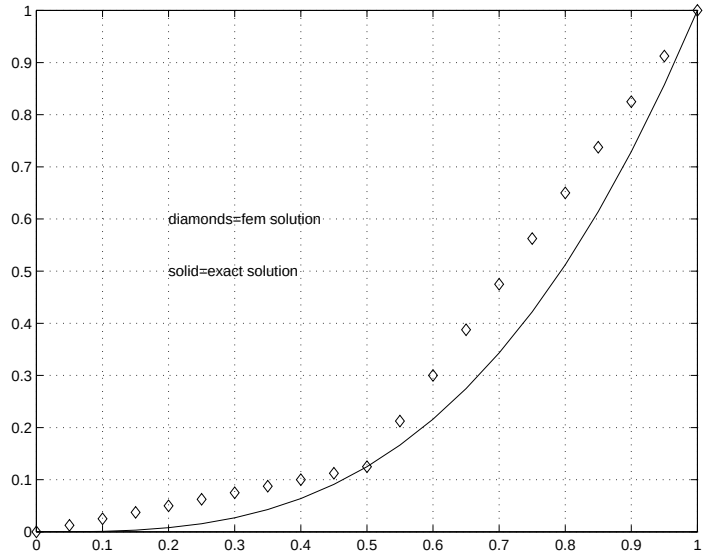

Figure 17: Comparison of exact and FEM approximate solution

$$\int_0^1 \left(\sum_{i=1}^{3} x_i \phi_i'(t) \right) \phi_2'(t) + \frac{3}{2} = 0$$

$$\sum_{i=\alpha}^{3} x_i \int_0^1 \phi_i'(t)\phi_2'(t)dt + \frac{3}{2} = 0 \Rightarrow \underbrace{x_1\left(-\frac{1}{h}\right) + x_2\left(\frac{2}{h}\right) + x_3\left(-\frac{1}{h}\right) + \frac{3}{2}}_{(\$)} = 0,$$

where $h = 1/2$. Substituting x_1 and x_3 gives $x_2 = \dfrac{1}{8}$ in (\$). We conclude that the approximation is

$$u(t) = 0.125\phi_2(t) + \phi_3(t)$$

A comparison of the exact and approximated solution appears in Figure (17).

Remark: One particularly nice feature of Galerkin/FEM and collocation methods is that the approximation $u(t)$ of the solution $Y(t)$ is defined over all of the range of t prescribed in the problem statement. This is not true for the finite difference solution, which only gives you an approximation y of Y, at specified locations t_i defined by the grid.

We'll consider more BVP issues in the context of PDE's, which is in the next part of the courseII.

□

0.2.8 Variational Formulation

Loosely based on Keller's "Boundary Value Problems" book.

A few nice properties of this analyses: (1) they force you to work out well-posedness issues along with some algorithmic issues simultaneously; (2) They often provide you with information on the space in which solutions exists and on the proper norm or measure of accuracy within that space; (3) in some problems, it allows you to analyze problems with weak smoothness requirements.

Here we'll only devote attention to a linear second-order differential equation with Dirichlet boundary conditions. This problem should serve as a good starting point for such methods as Finite Element and Boundary Element Methods, for Galerkin and collocation methods. The book by Claes Johnson on Finite Elements is a nice introduction to the finite element method (FEM); Canutto, Quarteroni, Hussaini, Zhang book on spectral methods is a good place to consult for spectral element methods.

Variational formulation for a second-order linear ODE

Assume problem is already in self-adjoint form (all linear 2nd-order ODE's can be cast in self-adjoint form):

$$\begin{cases} \mathcal{L}y = r(x) & a \le x \le b \\ y(a) = \alpha \\ y(b) = \beta \end{cases} \tag{100}$$

Where

$$\mathcal{L}y = -\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y \tag{101}$$

assume

$$(102) \qquad \begin{cases} p \in C^1[a,b] & q \text{ and } r \text{ continuous on } [a,b] \\ p(x) > 0, & q(x) > 0 \quad \text{on } [a,b] \end{cases}$$

Under these assumptions (100) has a unique solution.

We can exploit linearity to generate an equivalent problem with homogeneous boundary conditions:

Let $k(x)$ be a linear function having the same boundary values as y in (100). Let $z = y(x) - k(x)$. Then

$$(103) \qquad \begin{cases} \mathcal{L}z = r(x) - \mathcal{L}k(x) \\ z(a) = 0 \\ z(b) = 0 \end{cases}$$

is same type of problem as (100) but with homogeneous boundary conditions. By the way, this is a very useful trick to turn boundary value problems into inhomogeneous problem.

So we can solve (100) by solving (103). So we'll just consider

$$(104) \qquad \begin{cases} \mathcal{L}y = r(x) \quad a \le x \le b \\ y(a) = 0 \\ y(b) = 0 \end{cases}$$

Let the linear space $C_0^2[a,b] = \left\{ u \in C^2[a,b] \middle| u(a) = u(b) = 0 \right\}$. This defines the space of solutions.

So we write (104) as

$$\mathcal{L}y = r, \qquad y \quad \in C_0^2[a,b].$$

Note: $\mathcal{L} : C^2[a,b] \to C[a,b]$ is a linear operator. It's convenient to consider a slightly larger space:

$$V_0 = \left\{ v \in C[a,b] \middle| \begin{array}{l} v' \text{ is piecewise continuous and bounded on } [a,b] \\ \text{and } v(a) = v(b) = 0 \end{array} \right\}.$$

On V_0 we define the inner product

$$(u,v) \equiv \int_a^b u(x)v(x)dx \quad ; \quad v,u \in V_0. \tag{105}$$

Theorem: $\mathcal{L}$ in (101) is symmetric on $C_0^2[a,b]$ relative to the inner product (105), i.e.

$$(\mathcal{L}u, v) = (u, \mathcal{L}v) \quad \forall u, v \in C_0^2[a,b]$$

Proof: integration by parts:

$$\begin{aligned}
(\mathcal{L}u, v) &= \int_a^b \{[-p(x)u']' + q(x)u\}\, v(x)dx \\
&= -\left(pu'v\,\big|_a^b + \int_a^b [p(x)u'(x)v'(x) + q(x)u(x)v(x)]dx\right. \\
&= \int_a^b [p(x)u'(x)v'(x) + q(x)u(x), v(x)]dx
\end{aligned}$$

Since the last integral is symmetric in u and v, it is also equal to $(\mathcal{L}v, u)$, which in turn, by symmetry of $(\cdot,\cdot)$ proves the theorem.

□

The theorem's setting was on $C_0^2[a,b]$ but also works for V_0. It suggests an alternative inner product:

$$[u,v] \equiv \int_a^b [p(x)u'(x)v'(x) + q(x)u(x)v(x)]dx \quad u,v \in V_0$$

and the proof of the above theorem shows that

$$(\mathcal{L}u, v) = [u,v] \text{ if } u \in C_0^2[a,b], v \in V_0. \tag{106}$$

In particular, if $u = y$ is a solution of (104) then

$$[y,v] = (r,v) \quad , \quad \forall v \in V_0 \tag{107}$$

This is the "weak form" or "variational form" of (104).

Theorem: Let p^* be such that $p(x) \geq p^* > 0$. Under the assumptions made on p, q, r in (102), there exists positive constants c_1 and c_2 such that

$$c_1||u||_\infty^2 \leq [u,u] \leq c_2||u'||_\infty^2 \quad \forall u \in V_0. \tag{108}$$

Moreover,

$$c_1 = \frac{r^*}{b-a}, \quad c_2 = (b-a)||p||_\infty + (b-a)^3||q||_\infty.$$

Proof: For any $u \in V_0$, since $u(a) = 0 \Rightarrow$

$$u(x) = \int_a^x u'(t)dt, \quad x \in [a,b].$$

By Schwarz's inequality

$$u^2(x) \leq \int_a^x 1\, dt \int_a^x [u'(t)]^2 dt \leq (b-a)\int_a^b [u'(t)]^2 dt, x \in [a,b] \tag{109}$$

therefore

$$||u||_\infty^2 \leq (b-a)\int_a^b [u'(t)]^2 dt \leq (b-a)^2 ||u'||_\infty^2. \tag{110}$$

Using (102)

$$\begin{aligned} [u,u] = \int_a^b \{(p(x)[u'(x)]^2 + q(x)u^2(x)dx\} &\geq p^* \int_a^b [u'(t)]^2 dx \\ &\geq \frac{p^*}{b-a}||u||_\infty^2 \end{aligned}$$

where the last inequality follows from the left inequality in (110). This proves the lower bound in (108). The upper bound is found by observing that

$$[u,u] \leq (b-c)||p||_\infty ||u'||_\infty^2 + (b-c)||q||_\infty ||u||_\infty^2 \leq c_2 ||u'||_\infty^2,$$

where (110) has been used in the last step.

□

Remark: (108) implies uniqueness of solution of (104). In fact,

$$\mathcal{L}y = r \quad , \quad \mathcal{L}y^* = r \quad y, y^* \in C_0^2[a,b]$$

then $\mathcal{L}(y - y^*) = 0 \Rightarrow$ by (106) and (108):

$$\begin{aligned} &0 = (\mathcal{L}(y-y^*), y-y^*) = [y-y^*, y-y^*] \geq C, ||y-y^*||_\infty^2 \\ &\Rightarrow y = y^*. \end{aligned}$$

The Extremal Problem

Let

$$F(u) \equiv [u,u] - 2(r,u) \quad , \quad u \in V_0 \tag{111}$$

$F(u)$ is a quadratic functional. The extremal property for the solution y of (104) is stated in the following theorem:

Theorem: Let y be the solution of

$$\mathcal{L}y = r \quad y \in C_0^2[a,b].$$

Then $F(u) > F(y) \quad \forall u \in V_0 \quad u \neq y.$

Proof: By (107) $(r,u) = [y,u]$, so

$$\begin{aligned} F(u) &= [u,u] - 2(r,u) = [u,u] - 2[y,u] + [y,y] - [y,y] \\ &= [y-u, y-u] - [y,y] > -[y,y] \end{aligned}$$

where strict inequality holds by virtue of (109) and $y - u \neq 0$. On the other hand, since $[y,y] = (\mathcal{L}y, y) = (r,y)$, by (106):

$$F(y) = [y,y] - 2(r,y) = (r,y) - 2(r,y) = -(r,y) = -[y,y]$$

which combined with the other inequality proves the theorem.

□

This last theorem thus expresses the following extremal property of the solution of

$$\mathcal{L}y = r \quad y \in C_0^2[a,b] :$$

$$F(y) = \min_{u \in V_0} F(u) \tag{112}$$

We view (112) as an extremizing problem for determining y.

How do we find it? We'll do it on a machine. Thus, we'll solve (112) approximately by determining a function u_s from a finite dimensional subset $S \subset V_0$ that minimizes $F(u)$ on S.

A useful identity:

$$[y-u, y-u] = F(u) + [y,y] \quad , u \in V_0 \tag{113}$$

is satisfied by the solution y which was satisfied in the course of the proof of the last theorem.

Approximate Solution of Extremal Problem

Let $S \subset V_0$ be a finite-dimensional subspace of V_0 and $\dim S = n$. Let $u_1, u_2, \cdots, u_n$ be a basis for S, so that

$$u \in S \quad \rightarrow \quad u = \sum_{\nu=1}^{n} \eta_\nu u_\nu \quad \eta_\nu \in \mathbb{R} \tag{114}$$

We approximate y of (112) by $u_S \in S$, which satisfies

$$F(u_S) = \min_{u \in S} F(u). \tag{115}$$

Methodology:

For any $u \in S$,

$$\begin{aligned} F(u) &= \left[\sum_{\nu=1}^{n} \eta_\nu u_\nu, \sum_{\mu=1}^{n} \eta_\mu u_\mu \right] - 2\left(r, \sum_{\nu=1}^{n} \eta_\nu u_\nu \right) \\ &= \sum_{v,\mu=1}^{n} [u_\nu, u_\mu]\, \eta_\nu \eta_\mu - 2 \sum_{0=1}^{n} (r, u_\nu) \eta_v \end{aligned}$$

Let

$$M = \begin{pmatrix} [u_1,u_1] & [u_1,u_2] & \cdots & [u_1,u_n] \\ [u_2,u_1] & \vdots & \cdots & [u_2,u_n] \\ [u_3,u_1] & \vdots & \cdots & \vdots \\ \vdots & & \vdots & \vdots \\ [u_n,u_1] & [u_n,u_2] & \cdots & [u_n,u_n] \end{pmatrix}$$

this is often called the "Stiffness" matrix.

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{bmatrix}$$

the coefficient vector

$$f = \begin{bmatrix} (r, u_1) \\ (r, u_2) \\ \vdots \\ (r, u_n) \end{bmatrix}$$

the "Load" vector. In terms of these

$$F(u) = \eta^T M \eta - 2 f^T \eta \quad , \quad \eta \in \mathbb{R}^n$$

is the matrix form of the functional F.

M is symmetric positive definite, since $\eta^T M \eta = [u, u] > 0$ unless $u = 0$ (i.e $\eta = 0$).

So the extremal problem (115) takes the form

$$\begin{cases} \phi(\eta) = \min \\ \phi(\eta) \equiv \eta^T M \eta - 2 f^T \eta, \quad \eta \in \mathbb{R}^n \end{cases} \tag{116}$$

which is an unconstrained minimization on a quadratic in $\mathbb{R}^n$. Since M is positive definite then (116) has a unique solution $\hat{\eta}$ given by the solution of the linear system

$$M\eta = f \tag{117}$$

it is easily verified that

$$\phi(\eta) > \phi(\hat{\eta}) \forall \eta \in \mathbb{R}^n \quad \eta \neq \hat{\eta}; \tag{118}$$

Indeed, since $f = M\hat{\eta}$

$$\begin{aligned} \phi(\eta) &= \eta^T M \eta - 2 f^T \eta = \eta^T M \eta - 2 \hat{\eta} M \eta \\ &= \eta^T M \eta - 2 \hat{\eta} M \eta + \hat{\eta}^T M \hat{\eta} - \hat{\eta}^T M \eta \\ &= (\eta - \hat{\eta})^T M (\eta - \hat{\eta}) + \phi(\hat{\eta}) \end{aligned}$$

where $-\hat{\eta}^T M \hat{\eta} = -\hat{\eta}^T f = \hat{\eta}^T f - 2 f^T \hat{\eta} = \hat{\eta}^T M \hat{f} - 2 f^T \hat{\eta} = \phi(\hat{\eta})$ has been used in the last step. From this (118) follows.

Thus,

$$u_S = \sum_{\nu=1}^{n} \hat{\eta}_\nu u_\nu \quad , \quad \text{where } M\hat{\eta} = f. \tag{119}$$

In practice, the basis functions of S are chosen to have small support which results in the matrix M having a small bandwidth.

Next, we establish the *optimal approximation property* of u_S in the norm $[\cdot,\cdot]$, i.e.

$$[y-u_S, y-u_S] = \min_{u\in S}[y-u, y-n]. \tag{120}$$

By (113) and (115) the left-hand side $= F(u_S) + [y,y] = \min_{u\in S}\{F(u) + [y,y]\}$ which in turn equals the right-hand side of (120).

We can see that (120) leads to the following error estimate:

Theorem:

$$||y-u_S||_\infty \le \sqrt{c_2/c_1}\,||y'-u'||_\infty \quad \forall u \in S \tag{121}$$

where c_1 and c_2 are defined as

$$c_1 = \frac{p^*}{b-a} \quad c_2 = (b-a)||p||_\infty + (b-a)^3||q||_\infty. \tag{122}$$

In particular,

$$||y-u_S||_\infty \le \sqrt{c_2/c_1}\inf_{u\in S}||y'-u'||_\infty^2.$$

Proof:

By (122) and (120) we get

$$c_1||y-u_S||_\infty^2 \le [y-u_S, y-u_S] \le [y-u, y-u] \le c_2||y'-u'||_\infty^2$$

from which (121) follows. □

Remark: From above theorem we see that in order to get a good error bound we have to use an approximation process $y \approx u$, $u \in S$ which approximates the first derivative of y as well as possible. Note that this approximation process is independent of the one yielding u_S; its sole purpose is to provide a good error bound for U_s.

0.2.9 The Finite Element Method FEM

This is a specific Galerkin method, and as such the trial and test spaces are the same and the aim is to make the residual orthogonal to all elements in the space. First, some references. A good beginning book is Claes Johnson's "Numerical Solution of PDE's" (1987), other books are Phillippe Ciarlet's "FEM for Elliptic Problems" (1978) and Brenner and Scott's "FEM for PDE's" (1999). Most of these books emphasize the mathematical aspects. However, a lot of work has been put on the implementation side, which is a crucial component of the method and is unfortunately glossed over in these more mathematical books. Some good references on implementation issues are: .

As the name implies, finite elements use basis sets that are compactly supported. There are some nice advantages of FEM:

(1) mathematical- you are forced to think from the outset what function spaces you are using and in what sense, then, is a computed solution "close" to the exact solution, i.e. the spaces have built-in norms. A second mathematical aspect is that in casting the problem mathematically, one is forced to seriously consider whether the partial differential equation, or ordinary differential equation, is well-posed. That is not to say that you can use the analysis of Galerkin and the proceed to implement the approximate solution by some non-Galerkin technique.

(2) computationally - the technique allows us to obtain the solution everywhere in the domain, not just at grid locations. This, in contrast to a spectral (a special collocation case) or finite-difference technique, where you are approximating the solution at a finite set of grid locations. That is not to say that you can circumvent this issue by careful interpolation, but this is already provided by the Galerkin technique (and more importantly, you know an error estimate everywhere in the domain). Perhaps the most important advantage is that the technique lends itself very naturally to tiling very complex domains with elements that fit more naturally than simple uniform or rectangular lattices. It is thus very popular in boundary value problems (and eigenvalue problems) that come from very complex structures, such as bridges, buildings,structural components of vehicles, etc. This has been made considerably easier to do lately, with the availability of very smart automated

mesh generation packages (well, in 2d...but things are getting better in 3d), which take care of the most tedious and difficult part of the implementation of the method.

Again, here we focus only on elliptic problems with simple boundary conditions.

Dirichlet Model Problem Consider

$$\begin{aligned} -\mathrm{div}(a\,\mathrm{grad}u) &= f, \qquad \text{on } \Omega(x,y) \\ u &= 0, \qquad \text{on } \partial\Omega. \end{aligned} \tag{123}$$

Here $\Omega \subseteq \mathbb{R}^2$, $a, u, f : \Omega \to \mathbb{R}$. We assume that $0 \leq \underline{a} \leq a(x,y) \leq \overline{a}$. We call this the **Strong Formulation (SF)**.

A reminder:

$$\begin{aligned} \mathrm{grad}u &\equiv \frac{\partial u}{\partial x}\hat{x} + \frac{\partial u}{\partial y}\hat{y}, \\ -\mathrm{div}(a\,\mathrm{grad}u) &\equiv -\frac{\partial}{\partial x}(a\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(a\frac{\partial u}{\partial y}). \end{aligned}$$

Remark: note that when $a = 1$, we get the standard Poisson Equation problem we've studied before.

Weak Formulation (WF): Find u, vanishing on $\delta\Omega$, such that

$$\int_\Omega \mathrm{grad}u \cdot \mathrm{grad}v\, dxdy = \int_\Omega f v\, dxdy, \qquad \forall\, v \qquad \text{which vanish on } \delta\Omega. \tag{124}$$

Derivation: Use the divergence theorem (integration by parts in multi-dimensions). For $\mathbf{w}$ and v,

$$\int_\Omega (\mathrm{div}\mathbf{w}) v\, dxdy = -\int_\Omega \mathbf{w} \cdot \mathrm{grad}v\, dxdy + \int_{\delta\Omega} \mathbf{w} \cdot \hat{\mathbf{n}}\, v\, ds,$$

where $\hat{\mathbf{n}}$ is the outward normal to Ω

Let $\mathbf{w} = a\,\mathrm{grad}u$ yields (124), using $v = 0$ on $\delta\Omega$.

More precisely: for the **SF** we require $a(x,y)$ differentiable, and u twice differentiable, i.e. $a \in C^1$ and $u \in C^2$. For the **WF** we require a integrable and $\mathrm{grad}u \cdot \mathrm{grad}v$ integrable.

A little detour: we remind ourselves that a function space $L^2(\Omega)$ is comprised of the set of functions u for which

$$\int_\Omega |u|^2 d\Omega < \infty.$$

Example: Take the function $u = x^\beta$ and ask for what values of β is u an element of L^2 over the whole real line. If x and β are real,

$$\int_{-\infty}^{\infty} x^{2\beta} dx < \infty$$

if $\beta > -1/2$.

The *Sobolev Space* is a function space defined by the integrability of its elements. The Sobolev space $H^1(\Omega)$ is defined as

$$H^1(\Omega) = \left\{ u \in L^2(\Omega) | \qquad \partial_{x_i} u \in L^2(\Omega) \right\}$$

i.e. the function and all of its first derivatives are $L^2(\Omega)$. So, looking at spaces, $L^2 \supset H^1 \supset C^2 \supset C^1$. Another space, which we will make use of:

$$H_0^1(\Omega) = \left\{ u \in H^1(\Omega) | \qquad u = 0 \qquad \text{on } \delta\Omega \right\}$$

Theorem: If $u \in C^2$ and u satisfies the **SF** then u satisfies the **WF**. The proof is above.

Theorem: The **WF** has a unique solution.

The Variational Problem (V): Find $u \in H_0^1(\Omega)$ such that

$$E(u) = \min_{v \in H_0^1} E(v),$$

where

$$E(v) = \frac{1}{2} \int_\Omega a |\text{grad}\, v|^2 \, d\Omega - \int_\Omega f\, v \, d\Omega,$$

so that the functional $E : H_0^1 \to \mathbf{R}$.

Theorem u satisfies the **WF** if and only if u satisfies the variational problem.

Proof: suppose u satisfies the **WF**. Let's evaluate

$$E(v) = E(u + (v - u)) = \frac{1}{2}\int a|\text{grad}\, u + \text{grad}\,(v-u)|^2 - \int f(u + (v-u)).$$

Note that the integral sign assumes the integration is over the whole domain Ω. Expanding the first integral

$$E(v) = E(u) + \frac{1}{2}\int a|\text{grad}(v-u)|^2 + \int a\text{grad}u \cdot \text{grad}\,(v-u) - \int f(v-u).$$

The last two terms are zero since **WF** holds. Then

$$E(v) = E(u) + \frac{1}{2}\int a|\text{grad}\,(v-u)|^2 \geq E(u)$$

which implies the **VF**. □

Remark: For non-self-adjoint problems, can have **WF** but no **VF**. Note, however, that any linear second order **SF** problem can be put in self adjoint form (see Sturm-Liouville theory).

The problem

$$-\text{div} \cdot (a\text{grad}\, u) + \mathbf{b} \cdot \text{grad}\, u + cu = f$$

with zero Dirichlet boundary condtions has a **WF** but does not have a **VF** unless **b** is zero.

Exercise Let

$$b(u,v) = \int [a\,\text{grad}\, u \cdot \text{grad}\, v + \mathbf{b} \cdot (\text{grad}\, u)\, v + cuv]$$

Let $E(w) = \frac{1}{2}b(w,w) - F(w)$. Use $w = u + tv$, i.e. thinking of tv as a perturbation from u. Find the stationary condition for E about $t = 0$. What can you conclude from this calculation?

□

Theorem: If u satisfies the **VF**, then u satisfies the **WF**.

Proof: Choose any $v \in H_0^1$, and consider a real quantity t and $\rho : \mathbf{R} \to \mathbf{R}$, defined as

$$\rho(t) = E(u + tv) = \frac{1}{2}\int a|\text{grad}u + t\text{grad}v|^2 - \int f(u + tv)$$

if $\rho(0) = \min \rho(t)$, the $\rho'(0) = 0$.

$$0 = \rho'(0) = \int \operatorname{grad} u \cdot \operatorname{grad} v - \int f\, v,$$

which is the **WF**.

In what follows we will define

$$\begin{aligned} b(u,v) &\equiv \int a\mathrm{grad}u \cdot \mathrm{grad}v \\ F(v) &\equiv \int fv, \end{aligned}$$

so that $E(u) = \frac{1}{2}b(u,u) - F(u)$.

The numerical approximation of the **VF** problem is called the *Rayleigh-Ritz Method* and it can be summarized as follows: Choose a subspace $S_h \in H_0^1$, where S_h is finite dimensional subspace. Then, finding the $u_h \in S_h$ that minimizes $E(v)$ yields an approximation to $u_h \approx u$, if S_h is "sufficiently large." This could be called the $\mathbf{VF}_h$ problem.

The numerical approximation of the **WF** problem can be done using Galerkin techniques. Briefly, use a finite dimensional set of basis for S_h. Supposing $\dim S_h = N$, then, every function v in S_h can be written as

$$v = \sum_{i=1}^{N} \alpha_i \phi_i.$$

where ϕ are the bases. Substituting in **WF** gives

$$b(u_h, v) = F(v)$$

or

$$b(\sum_{i=1}^{N} \alpha_i \phi_i, \phi_j) = F(\phi_j) \qquad i,j = 1,2,\ldots N$$

which is thus

$$\sum_{i=1}^{N} \alpha_i b(\phi_i, \phi_j) = F(\phi_j) \qquad i,j = 1,2,\ldots N.$$

$b(\phi_i, \phi_j)$ with $i, j = 1, 2, \ldots N$ is a matrix, we shall call it the *stiffness matrix* M, and $F(\phi_j)$, with $j = 1, 2, \ldots N$ is a vector we shall call the *load vector* $\mathbf{F}$. So in matrix notation, the task is to solve

$$M\alpha = \mathbf{F}$$

for the unknown vector α.

What happens if the problem is instead nonlinear? for example, suppose we want to solve

$$-\text{div} \cdot (a(x,u)\text{grad}\, u(x)) = f(x) \qquad x \in \Omega$$

subject to the zero Dirichlet boundary conditions, with the usual assumptions on $a(x,u)$. In that case we are lead to a nonlinear set of equations, given by

$$\int a(x, \sum_j \alpha_j \phi_j)\text{grad}\, \phi_j \cdot \text{grad}\, \phi_i = \int f\phi_i$$

with $i, j = 1, 2, \ldots, N$. We know how to solve these types of systems using a root finding algorithm (For Newton's method, see).

Exercise Set up the **WF** problem and its Galerkin approximation for

$$-u''(x) = f \qquad 0 < x < 1, \tag{125}$$

with boundary conditions $u(0) = u(1) = 0$. Let $S_h = \mathcal{P}^9$, where the basis set is $\phi_i = x^i(1-x)$, $i = 0, 1, \ldots 9$. Prove that the underlying linear algebraic problem has a stiffness matrix that is symmetric and nonsingular.

□

Model Problem with Neumann Boundary Conditions

$$-\text{div}\,(a\text{grad}\, u) = f \qquad \text{on } \Omega \tag{126}$$

$$a\frac{\partial u}{\partial n} = 0 \qquad \text{on } \partial\Omega \tag{127}$$

If $u \in H^m$ then the partial derivatives up to order $m-1$ are defined on $\partial\Omega$ and are L^2 functions there. However, partials of order m cannot be sensibly

defined on $\partial\Omega$. So to define a space of functions that is H^1 such that $a\partial/\partial n$ of these functions are zero on the boundary is nonsensical.

Instead, let $v \in H^1$.We show that this is a good function space for the trial functions:

$$-\int_\Omega \mathrm{div} \cdot (a\mathrm{grad}\, u)v = \int_\Omega a\mathrm{grad}\, u \cdot \mathrm{grad}\, v - \int_{\partial\Omega} (a\mathrm{grad}\, u) \cdot \hat{n}v = \int_\Omega fv$$

which is

$$\int_\Omega a\mathrm{grad}\, u \cdot \mathrm{grad}\, v - \int_{\partial\Omega} a\frac{\partial u}{\partial n} = \int_\Omega fv.$$

If u solves (127), then the second term on the left hand side is zero. Hence, the **WF** of the problem is

$$\int_\Omega a\mathrm{grad}\, u \cdot \mathrm{grad}\, v = \int_\Omega fv, \qquad \forall v \in H^1.$$

Remark: we say that $a\frac{\partial u}{\partial n} = 0$ is a *natural boundary condition*, whereas $u = 0$ is an *essential boundary condition*.

Exercise Set up the **WF** problem and its Galerkin approximation for

$$\begin{aligned} -u''(x) + au'(x) + bu(x) &= f \qquad 0 < x < 1, \\ u'(x) &= c_0 \qquad x = 0 \\ u'(x) &= c_1 \qquad x = 1 \end{aligned}$$

(128)

Let $S_h \subset H^1$ be the set of N hat piece-wise linear functions defined on the distinct nodes x_i, $i = 1, 2, \ldots N$ defined on the interval $0 \le x \le 1$. Note: you should get that c_1 and c_2 make contributions to the load vector.

□

An Error Estimate: for the problem posed in (125): take $S_h \subset H_0^1$, then

$$b(u_h, v) = (f, v) \qquad \forall v \in S_h.$$

and

$$b(u, v) = (f, v) \qquad \forall v \in S_h.$$

where

$$u_h = \sum_{i=1}^{N} \alpha_i \phi_i$$

where the set $\{\phi_i\}_{i=1}^N$ span S_h. Then

$$b(u - u_h, v) = 0 \qquad \forall v \in S_h.$$

Theorem For any $v \in S_h$, as above,

$$||(u - u_h)'|| \leq ||(u - v)'||.$$

Proof: Let $v, u_h \in S_h$, and $w = u_h - v$. Then

$$\begin{aligned} ||(u - u_h)'||^2 &= b(u - u_h, u - u_h) + b(u - u_h, w) \\ &= b(u - u_h, u - u_h + w) = b(u - u_h, u - v) \leq ||(u - u_h)'||||(u - v)'|| \end{aligned}$$

by Cauchy Schwarz. □

Furthermore, if ϕ_i are piecelinear hat functions,

$$|u - u_h| \leq \frac{h^2}{8} \max_{y \in \Omega} |u''(y)|$$

where h is the largest distance between the nodes. This is the same estimate you would get for the finite difference approximation of the equation using center difference approximation for the second derivative.

□

Worked Two-Dimensional Example Calculation Here we will work through a two-dimensional calculation. We will assume that you will be doing all of the work. We'll provide you with some answers for what you should get along the way.

We'll solve

$$\begin{aligned} \nabla^2 u = f(x, y) \qquad & (x, y) \in \Omega \\ u(x, y) = \alpha(x, y) \qquad & (x, y) \in \partial\Omega_1 \\ \frac{\partial u}{\partial n(x, y)} = \beta(x, y) \qquad & (x, y) \in \partial\Omega_2 \end{aligned}$$

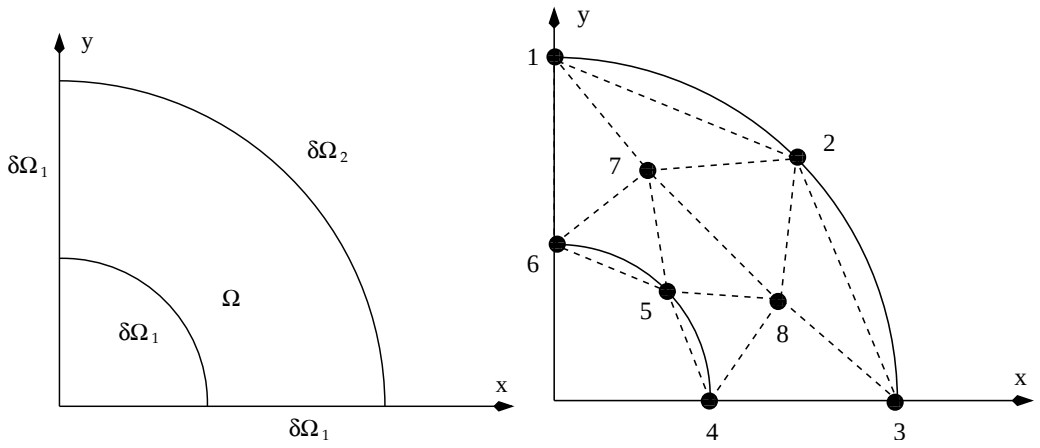

Figure 18: (a) Domain for worked example; (b) node and element assignment.

See Figure 18.

We will assume in this example that $f = 0$, $\alpha = 0$, and $\beta(\theta) = \sin 2\theta$. We will take the inner radius to be $r = 1$ and the outer one to be $r = 2$. Here $x = r\cos\theta$, and $y = r\sin\theta$. The Laplacian in polar coordinates, applied to some function w, reads

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 w}{\partial \theta^2}.$$

We will tile the domain with 8 triangular elements. Each node point in the figure is numbered.

To label node points we will use a local and a global scheme. A single index will denote a node in the global scheme as given in the figure. A double index will denote the local scheme. In this local scheme the node points are numbered $k1$, $k2$, $k3$ going in counterclockwise direction where k is the element number. For example, in element V the local nodes 51, 52, 53 correspond to the global nodes 2, 7, 8, respectively. We will use piecewise bilinear elements. hence our approximation will have C^0 global continuity.

Construction of the elements Let

$$\phi_{ki}(x, y) = \gamma_1^i + \gamma_2^i x + \gamma_3^i y$$
$$\phi_{ki}(x_{ki}, y_{ki}) = 1 \qquad \phi_{ki}(x_{kj}, y_{kj}) = 0 \quad i \neq j$$

Take $i = 1$

$$\phi_{k1}(x_1, y_1) = \gamma_1^1 + \gamma_2^1 x_1 + \gamma_3^1 y_1 = 1$$
$$\phi_{k1}(x_2, y_2) = \gamma_1^1 + \gamma_2^1 x_2 + \gamma_3^1 y_2 = 0$$
$$\phi_{k1}(x_3, y_3) = \gamma_1^1 + \gamma_2^1 x_3 + \gamma_3^1 y_3 = 0$$

Solving gives

$$\gamma_1^1 = \frac{x_2 y_3 - x_3 y_2}{2A}$$
$$\gamma_1^2 = \frac{y_2 - y_3}{2A}$$
$$\gamma_1^3 = \frac{x_3 - x_2}{2A}$$

where

$$A = \frac{1}{2} \det \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

i.e. the area of the triangle.

The other two basis functions that are nonzero in the k–th element are

$$\phi_{ki}(x, y) = \frac{1}{2A}\left[(x_{k(i+1)}y_{k(i+2)} - x_{k(i+2)}y_{k(i+1)} + (y_{k(i+1)} - y_{k(i+2)})x + (x_{k(i+2)} - x_{k(i+1)})y\right]$$

where $i = 2, 3$ and the indicial operations are taken mod 3 with counterclockwise local numbering assumed.

We want an approximation of the solution in the form

$$v(x, y) = \sum_{j=1}^{N} v_j \phi_j(x, y) \tag{129}$$

where N is the number of node points. Again we multiply the given differential equation by ϕ_i and integrate. This way we get

$$\int_\Omega \nabla^2 v \phi_i(x, y)\, \mathrm{d}x\, \mathrm{d}y = \int_\Omega f \phi_i(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{130}$$

Integration by parts produces

$$\int_\Omega \nabla^2 v \phi_i(x, y)\, \mathrm{d}x\, \mathrm{d}y = \int_{\partial\Omega} (\nabla v \cdot \hat{n})\, \mathrm{d}s - \int_\Omega \nabla v \nabla \phi_i\, \mathrm{d}x\, \mathrm{d}y \tag{131}$$

where $\hat{n}$ is the outward unit normal on each segment of $\partial\Omega$ and $\hat{n}\cdot\nabla v = \frac{\partial v}{\partial n}$.

Using (129) and (131) we rewrite (130) as follows

$$\int_{\partial\Omega} \frac{\partial v}{\partial n}\phi_i \,\mathrm{d}s - \sum_{j=1}^{N} v_j \int_{\Omega} \nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}x\,\mathrm{d}y = \int_{\Omega} f\phi_i \,\mathrm{d}x\,\mathrm{d}y \tag{132}$$

since $f = 0$, the RHS vanishes. The

$$\int_{\Omega} \nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}x\,\mathrm{d}y$$

terms give entries into what is commonly called the stiffness matrix M, a $N \times N$ square matrix. The quantities

$$\int_{\partial\Omega} \frac{\partial v}{\partial n}\phi_i \,\mathrm{d}s - \int_{\Omega} f\phi_i \,\mathrm{d}x\,\mathrm{d}y$$

are called load vector F. So if $V = [v_1, v_2, \ldots, v_N]^T$ then (132) is nothing more than

$$MV = F$$

We most efficiently compute the stiffness matrix by using the relation

$$\int_{\Omega} = \sum \int_{\Omega_k}$$

where Ω_k is the domain of each element. The plan is finding the contributions for the stiffness matrix from every element and then adding up all these contributions. So we need to evaluate

$$\int_{\Omega^e} \nabla\phi_l \cdot \nabla\phi_m \,\mathrm{d}x\,\mathrm{d}y = \int_{\Omega} \left(\frac{\partial\phi_l}{\nabla x}\frac{\partial\phi_m}{\nabla x} + \frac{\partial\phi_l}{\nabla y}\frac{\partial\phi_m}{\nabla y} \right)$$

$l, m = 1, 2, 3$. You should be able to show that

$$\int_{\Omega^e} \nabla\phi_l \cdot \nabla\phi_m \,\mathrm{d}x\,\mathrm{d}y =$$
$$= \frac{1}{4A}\Big[(y_{l+1} - y_{l+2})(y_{m+1} - y_{m+2}) + (x_{l+2} - x_{l+1})(x_{m+2} - x_{m+1})\Big]$$

The load vector $\int \partial v/\partial n \,\mathrm{d}s$ can be found as follows: It is easy to see that it vanishes for all interior nodes because those ϕ_i are zero along the boundary

$\partial\Omega$. Let us look at node 2. Assume that $\partial v/\partial n = 1$ and use the fact that ϕ_2 is linear on $\partial\Omega$. Then

$$\begin{aligned}\int \frac{\partial v}{\partial n}\phi_2 \,\mathrm{d}s &= \int_{\text{node 1}}^{\text{node 2}} 1\cdot\phi_8 \,\mathrm{d}s + \int_{\text{node 2}}^{\text{node 3}} 1\cdot\phi_8 \,\mathrm{d}s \\ &= \int_0^{l_{1-2}} \frac{s}{l_{1-2}} \,\mathrm{d}s + \int_0^{l_{2-3}} \frac{s}{l_{2-3}} \,\mathrm{d}s \\ &= \frac{l_{1-2}}{2} + \frac{l_{2-3}}{2}\end{aligned}$$

where l_{i-j} denotes the distance from node i to node j.

To solve the given problem, follow these steps:

1. Find the coordinates of all the node points.

node number	coordinates
1	(0, 2)
2	(1.4142, 1.4142)
3	(2, 0)
4	(1, 0)
5	(0.7071, 0.7071)
6	(0, 1)
7	(0.75, 1.299)
8	(1.299, 0.75)

2. Make a table assigning to each element its corresponding node points in counterclockwise motion.

element number	nodes
I	1,7,2
II	1,6,7
III	6,5,7
IV	2,7,8
V	5,8,7
VI	3,2,8
VII	5,4,8
VIII	4,3,8

3. Find the local stiffness matrices for each element. Use symmetries and a matlab script to make the task less tedious.

   As all stiffness matrices are symmetric only the upperdiagonal part is given. Element I

0.4116	-0.7897	0.3781
	2.1224	-1.3327
		0.9546

Element II

0.4346	-0.2353	-0.1993
	0.7026	-0.4673
		0.6667

Element III

0.4086	-0.2426	-0.1659
	0.7563	-0.5136
		0.6796

Element IV

0.7045	-0.3523	-0.3523
	0.5311	-0.1789
		0.5311

Element V

0.8646	-0.4323	-0.4323
	0.5051	-0.0728
		0.5051

Element VI

0.4116	0.3781	-0.7897
	0.9546	-1.3327
		2.1224

Element VII

0.7563	-0.2426	-0.5136
	0.4086	-0.1659
		0.6796

Element VIII

0.7026	-0.2353	-0.4673
	0.4346	-0.1993
		0.6667

4. Add up the element stiffness matrices to obtain the global stiffness matrix.

0.8462	0.3781	0	0	0	-0.2353	0.989	0
	2.6137	0.3781	0	0	0	-1.685	-1.685
		0.8462	-0.2353	0	0	0	0.989
			1.1112	-0.2426	0	0	-0.6332
				2.3772	-0.2426	-0.9459	-0.9549
					1.1112	-0.6332	0
						4.5059	-0.2517
							4.5049

You should try to do this on your own. Download a possible solution to this computational task by clicking here

e) Find the force vector.

The only contribution comes from the second node. We will integrate the boundary terms over the arc of the circle. If many points on the arc are used the difference between integration along the arc to integration along the triangle edges is small and the integrals are much more convenient along the arcs. Let us just consider element VI which will give us $F_2/2$ because of symmetry. Using polar coordinates we have $\mathrm{d}s = 2\,\mathrm{d}\theta$ and we can interpolate ϕ_2 linearly which will gives us

$$\frac{F_2}{2} = \int_{\text{node 2}}^{\text{node 3}} 1 \cdot \phi_2 \,\mathrm{d}\theta = 2\int_0^{\frac{\pi}{4}} \Big(\frac{\pi}{4} - \theta\Big) \sin(2\theta)\,\mathrm{d}\theta$$

5. Impose the Dirichlet boundary conditons.

   All nodes except 2, 7, 8 have zero function value. Therefore the stiffness matrix reduces to a 3×3 matrix and we end up with the following system of equations:

$$\begin{pmatrix} 2.6137 & -1.685 & -1.685 \\ -1.685 & 4.5049 & -0.2517 \\ -1.685 & -0.2517 & 4.5049 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0.5708 \\ 0 \\ 0 \end{pmatrix}$$

6. Solve the resulting system of equations.

   We obtain $v_1 = 0.4464$, $v_2 = 0.1769$, $v_3 = 0.1769$.

7. Compare the results with the exact solution. The exact solution is $u(r, \theta) = 4/17(r^2 - 1/r^2)\sin 2\theta$.

Our approximation is, not surprisingly, poor. However, only three free nodes were included in the calculation. More elements are needed for a better result. □

□

Part II

PARTIAL DIFFERENTIAL EQUATIONS (PDE's)

References Used:

Strikwerda "Finite Difference schemes and PDE'S"
Iserles "A first Course in the Numerical Analysis of Differential Equations"
LeVeque "Numerical Methods for Conservation Laws"
Richtmyer and Morton "Difference Methods for IVP"
Hall and Persching "Numerical Analysis of PDE's"
McCormick: "Tutorial on Multigrid"

Intro books on Spectral and FEM

Orszag and Gottlieb: "Tutorial on spectral methods",
Cannuto, et al. "Spectral Methods in Fluid Dynamics"
C. Johnson: "Intro to FEM".
Strang and Fixx: "Intro to FEM"

0.3 INTRODUCTION

0.3.1 Basic Methods for Numerical Approximation of PDE

1. Finite Difference Techniques (we'll concentrate on these) $\equiv FD$
2. Spectral Methods $\equiv SM$
3. FEM/Galerkin $\equiv FEM$

CLASSIFICATION OF PDE'S

PDE's come in all shapes and forms. Each type, you will find, will require a different numerical approximating method. We will concentrate on three types of problems, which are ubiquitous in physics and engineering

Type Classification	Canonical Example
Hyperbolic	Wave equation $U_t + AU_x = 0$
Parabolic	heat equation $U_t = kU_{xx}$
Elliptic	Poisson equation $\nabla^2 U = f$

We will also consider an equation of mixed type, the "Advection-Diffusion equation."

Type Classification: Within our purview the type classification is not a tremendously important concern. Perhaps more useful is to make the association between the names and the canonical examples. Nevertheless the type name comes from the classification of linear second-order pde's. Take $u = u(x, y)$ and a, b, c, d, e, f real constants. The equation

$$au_{xx} + bu_{xy} + cu_{yy} + du_x + eu_y + fu = g$$

is

$$\begin{aligned} &\text{elliptic if } b^2 - 4ac < 0 \\ &\text{parabolic if } b^2 - 4ac = 0 \\ &\text{hyperbolic if } b^2 - 4ac > 0 \end{aligned}$$

The names come from an analogy with conic sections. Consult a PDE book for more details.

BASIC PROBLEM

Every PDE type equation requires a special or particular strategy of numerical approximation. Traditionally courses on numerical methods for PDE's have been organized by method (i.e. FD, SM, FEM, etc.) rather than by equation type. The reasons have to do with the fact that this relatively young subject has been taught by researchers who specialize in the methods rather than in the equation types.

We will follow tradition here: we will cover mostly the FD method of mostly linear hyperbolic, parabolic, and elliptic equations.

Roughly speaking we can divide the PDE families into

- Evolutionary (causal or "time" dependent problems for example the heat equation (Parabolic), the wave equation (Hyperbolic): IV/BV Problems
- Non-Evolutionary BVP, for example Poisson's equation (elliptic)

In many physical situations one might also find equations of mixed type, or problems that couple equations of various types.

0.4 HYPERBOLIC EQUATIONS

Perhaps the hardest evolutionary PDE problems to approximate are hyperbolic. Reasons for this are beyond scope of the course, but you'll be getting an appreciation of the difficulty in the homework assignments. Why start with the most difficult PDE's? Because they serve as a good venue to illustrate some of the basic numberical-analytical concepts. What we will do is solve some hyperbolic problems and avoid most of the complicated aspects.

In what follows we'll denote x the spatial variable, which can be the whole real line or a closed subinterval of the real line. We will denote time as t and assume $t \geq 0$. The dynamic variable is $U = U(x,t)$

Three arquetypical problems that are Hyperbolic, are

$U_t + A(x,U,t)U_x = 0$	Advection or One-Way Wave Equation
$U_{tt} - \frac{1}{c^2}U_{xx} = 0$	$c = c(x,U)$ The Wave Equation
$U_t + [f(U)]_x = 0$	A Conservation law. Here f doesn't depend on U_x, U_{xx}, etc...

Let's take a look at the scalar One-way Wave Equation: Take $U = U(x,t) \in \mathbb{R}^1$

The Simple Advection Equation

$$U_t + aU_x = 0 \tag{133}$$

a, constant

$$U(x,0) = U_0(x)$$

has solution $U_0(x-at)$, that is,

$$U(x,t) = U_0(x-at) \tag{134}$$

See Figure 19 for an illustration. Note that the wave travels to the right as time increases and the shape is retained.

What we glean from solution:

1. requires differentiability of U, but (134) does not, makes sense ... this introduces the concept of "weak" solutions (e.g. shocks). Hyperbolic problems admit solutions with discontinuities .

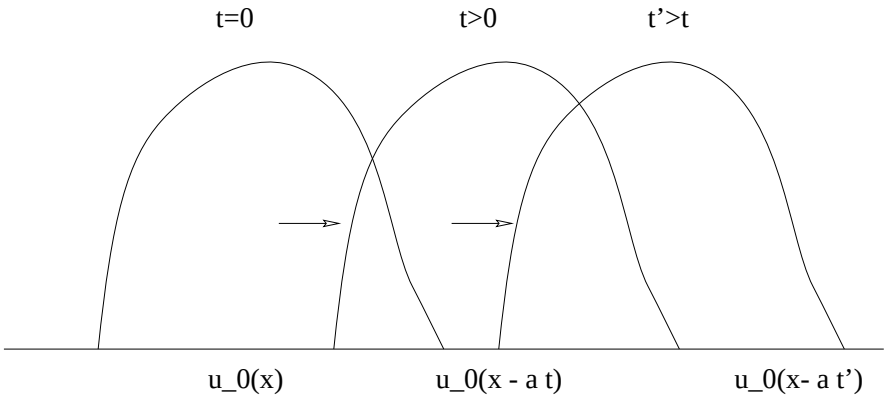

Figure 19: Graphical representation of the solution to the simple advection or one way wave equation with constant speed $a > 0$

2. See solution is a copy of U_0 except displaced by at (to the right if $a > 0$, to the left if $a < 0$).

3. Solution only depends on *characteristic* variable $\xi = x - at$.

Solution to a more general advection equation

$$\text{(135)} \qquad \begin{cases} U_t + aU_x + bU = f(t,x) & x \in \mathbb{R}^1, t > 0 \\ U(0,x) = U_0(x) \end{cases}$$

a, b constants

$$\begin{cases} \tau = t \\ \xi = x - at \end{cases} \Rightarrow \begin{cases} t = \tau \\ x = \xi + a\tau \end{cases}$$

and define $\tilde{U}(t,\xi) = U(t,x)$ (same function in both coordinate systems).

$$\therefore \frac{\partial \tilde{U}}{\partial \tau} = \frac{\partial t}{\partial \tau}\frac{\partial U}{\partial t} + \frac{dx}{d\tau}\frac{\partial U}{\partial x} = \frac{\partial U}{dt} + a\frac{\partial U}{dx} = -bU + f(t, \xi + a\tau)$$

$\Rightarrow$ We get an ODE:

$\frac{\partial \tilde{U}}{\partial \tau} = -bU + f(\tau, \xi + a\tau)$ which we can solve:

$$\begin{aligned} \tilde{U}(t,\xi) &= U_0(\xi)e^{-b\tau} + \int_0^\tau f(\sigma, \xi + a\sigma)e^{-b(\tau-\sigma)}d^\sigma \\ \therefore \quad U(t,x) &= U_0(x-at)e^{-bt} + \int_0^t f(s, x - a(t-s))e^{-b(t-s)}ds \end{aligned}$$

What we learn from this solution:

- Solution is found by a change of coordinate system, or more precisely, a change of reference frame.
- Along special lines in space-time, the function U is an ODE. These lines are called *characteristics*. In this instance the lines are straight, but they are not in general, if the speed a is not constant in space or time.
- The resulting ODE above shows that if $b > 0$ that $\tilde{U}$ dissipates and grows if $b < 0$. Note that if b and f is the derivative of another function that we get a conservation law for $\tilde{U}$, and that $\tilde{U}$ is conserved if $f = 0$.

Equations with Variable Coefficients Now assume that the speed $a = a(x,t)$. Then

$$\left\{ \begin{array}{c} U_t + a(t,x)U_x = 0 \\ U(0,x) = U_0(x) \end{array} \right. \tag{136}$$

then

$$\begin{aligned} \frac{\partial \tilde{U}}{\partial \tau} &= \frac{\partial t}{\partial \tau} U_t + \frac{\partial x}{\partial \tau} U_x = 0 \\ &= U_t + a U_x = 0 \end{aligned}$$

$$\therefore \quad \frac{dx}{d\tau} = a(t,x) = a(\tau, x)$$

$\therefore$ (136) is equivalent to

$$\left\{ \begin{array}{ll} \dfrac{d\tilde{u}}{\partial \tau} = 0 & \tilde{U}(0,\xi) = U_0(\xi) \\ \dfrac{dx}{d\tau} = a(\tau, x) & x(0) = \xi \end{array} \right.$$

Example)

$$U_t + xU_x = 0$$

$$U(0,x) = \left\{ \begin{array}{ll} 1 & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{array} \right.$$

$$\Rightarrow \frac{d\tilde{U}}{d\tau} = 0,\ \frac{dx}{d\tau} = x, \quad x(0) = \xi$$

the second of these can be integrated to give $x(t) = ce^{\tau}$
Thus,

$$x(0) = \xi = c \quad \therefore \quad x(\tau) = \xi e^{\tau}$$
$$\therefore \xi = xe^{-t}$$
$$\therefore \tilde{u} = \tilde{u}(\xi)$$

$$\tilde{u}(t, \xi) = u_0(\xi)$$
$$\therefore \quad u(t, x) = \tilde{u}(t, \xi) = u_0(\xi) = u_0(xe^{-t})$$

so we get, for $t > 0$

$$U(t, x) = \begin{cases} 1 & \text{if } 0 \le x \le e^t \\ 0 & \text{otherwise.} \end{cases}$$

Quasi-linear (mildly nonlinear) Equations Consider

$$U_t + UU_x = 0$$

with

$$U(0, x) \equiv \phi(x) = \begin{cases} 2 & x < 0 \\ 2 - x & 0 \le x \le 1 \\ 1 & x > 1 \end{cases}$$

Since the speed $a(u) = U(x, t)$, the characteristics are straight lines emanating from $(\xi, 0)$ with speed $a(\phi(\xi)) = \phi(\xi)$. For $x < 0$ the lines have speed 2. for $x > 1$ the lines have speed 1. For $x \in [0, 1]$ the lines have speed $2 - x$ and these all intersect at $x = 2$ and $t = 1$. Thus, solution cannot exist for $t > 1$. Actually, it does in what we call *weak* form. At $t = 1$ we get wave breaking or a shock, i.e. the function no longer is single valued. To find the solution for $t < 1$ note $U(x, t) = 2$ for $x < 2t$ and $U(x, t) = 1$ for $x > t + 1$. For $2t < x < t + 1$ we get

$$x = (2 - \xi)t + \xi$$

which in turns gives

$$\xi = \frac{x - 2t}{1 - t}$$

Thus the solution is

$$U(x,t) = \frac{2-x}{1-t}$$

for $2t < x < t+1$, and $t < 1$.

Remarks: As we see from the above examples hyperbolic problems propagate signals or information -in the form of waves- with finite speed. An example of such information is the initial data. The direction in which the signal travels depends on the sign of the speed: as posed above, and for $t > 0$, the signal will travel at speed $|a|$ and to the right if $a > 0$ (remember that this speed may depend on x, t, even on U), and to the left if $a < 0$ (see Figure 19).

Let's consider a system of hyperbolic equations:

$$\begin{pmatrix} U \\ V \end{pmatrix}_t + \begin{pmatrix} a & b \\ b & a \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix}_x = 0$$

$$0 \le x \le 1$$

the eigenvalues or eigenspeeds are $a+b, a-b$. Take $a, b > 0$, constants.

If $0 < b < a \Rightarrow$ both characteristics travel to right.
$0 < a < b \Rightarrow$ charactertistics travel in opposite direction

The situation is portrayed in Figure 20

Physical problems are often posed on a finite span in x. Assume this span is of length l. The hyperbolic problems considered above are well posed if initial data is specified and appropriate boundary conditions used, and all of these are consistent. Not only is information from the initial data advected but so is boundary data that is to the left (right) and before if $a > 0$ (if $a < 0$). One of the many difficulties associated with hyperbolic problems is in fact the issue of boundary conditions. Since we are always computing over finite domains, they will always require careful consideration. In what follows of this presentation we will not consider the hard boundary issues....even in your assignments these will be carefully avoided.

By way of example
Consider

$$\begin{aligned} U_t + aU_x &= 0 \\ 0 \le x \le 1 \quad t > 0 & \end{aligned}$$

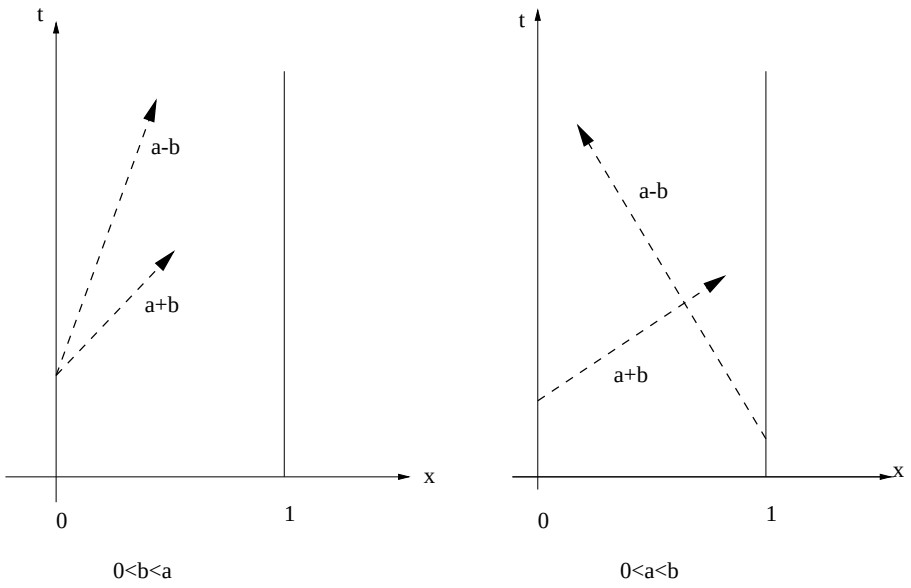

Figure 20: Signal propagation direction, depending on the size of the eigenspeeds

and

take $a > 0$ $\begin{cases} U(0,x) = U_0(x) \\ U(t,0) = g(t) \end{cases}$

then solution is $\begin{cases} U_0(x-at) & x-at > 0 \\ g(t-a^{-1}x) & x-at < 0 \end{cases}$

Along $x - at = 0$ there'll be a jump in the solution if $u_0(0) \neq g(0)$.

For $a < 0$, roles are reversed. See Figure 21 (Convince yourself!)

Periodic Boundary Conditions: in this case we prescribe $u(t, x+l) = u(t, x)$, where l is length of strip. These can add strong structure to solution.

Finite Difference Schemes

Most of this material comes from Rychtmyer and Morton's monograph and Strikwerda's textbook.

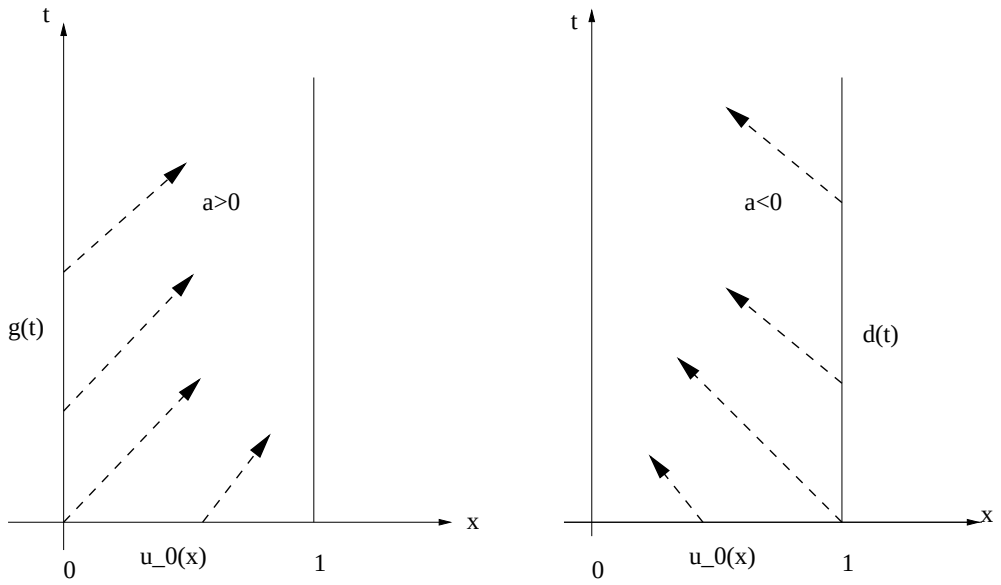

Figure 21: Data specification for $a > 0$ and $a < 0$ case

Take domain (x, t) and define a lattice $(x_m, t_n) = (mh, nk)$, where m and n are integers. Typically, $n \geq 0$. We limit the presentation to grids which are uniform in both x and t (see Figure 22).

here $\begin{cases} h \text{ is } x \text{ grid spacing} \\ k \text{ is } t \text{ grid spacing} \end{cases}$

Notation: Let $U_m^n = U(x_m, t_n) = U(mh, nk)$ be the value of U on the lattice. Let u_m^n be an approximation of U_m^n on the same lattice location. We're going to consider mostly grids with constant grid spacing.

As in the ODE case, the most important properties of any numerical scheme for the approximation of a PDE (not just hyperbolic ones) are:

- Convergence
- Consistency
- Stability

Definition: **Convergence:** for one-step schemes approximating a ANY PDE to be convergent we compare $U(x, t)$ and u_m^n: if U_m^0 converges to $U_0(x)$ as

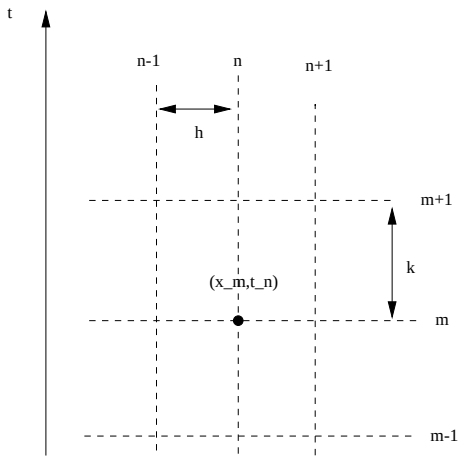

Figure 22: Space-time Lattice about x_m, t_n

$mh \to x$ then U_m^n converges to $U(x,t)$ at (m_h, nk) converges to (x,t) as $h,k \to 0$. As $h,k \to 0$ the approximation gets uniformly closer to exact solution on the lattice.

Some properties of a scheme that we should be interested in are:

- Order of Accuracy
- Dissipation, Dispersion
- Speed and Efficiency

Of course, this list of properties is not exhaustive and the properties of importance are different for different types of PDE's.

A fundamental theorem of Finite Difference approximations of PDE's is the **Lax-Richtmyer Equivalence Theorem**

THE LAX-RICHTMYER EQUIVALENCE THEOREM

A consistent finite difference scheme for a PDE for which the initial value problem is well-posed is convergent if and only if it's stable.

Proof: See Chapter 10 Strikwerda's book.

So while proof of convergence would be a function-analytic exercise, we could instead check for consistency and stability and get convergence as a bonus. This is nice since stability and consistency is usually easier to check than convergence.

What's consistency?

definition:

Given a PDE of the form $PU = f$ and finite difference scheme $P_{k,h}u = f$, we say the FD scheme is consistent with the PDE if, for any smooth $\phi(t,x)$,

$$P\phi - P_{k_1,h}\phi \to 0 \quad \text{as } h,k \to 0,$$

the convergence being pointwise convergence at each grid point.

Basic idea in Finite Difference Methods: replace derivatives by finite difference approximations. What we obtain is a pointwise approximation on a grid (no information on points not belonging to the lattice)

For the equation

$$U_t + a\,U_x = 0$$

some schemes are

$$\text{"forward space-forward time"} \quad \frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_{m+1}^n - u_m^n}{h} = 0$$

$$\text{"forward time-centered space"} \quad \frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_{m+1}^n - u_{m-1}^n}{2h} = 0$$

$$\text{"leapfrog"} \quad \frac{u_m^{n+1} - u_m^{n-1}}{2k} + a\frac{u_{m+1}^n - u_{m-1}^n}{2h} = 0$$

$$\text{"Lax-Friedrichs"} \quad \frac{u_m^{n+1} - \frac{1}{2}\left(u_{m+1}^n + u_{m-1}^n\right)}{k} + a\frac{u_{m+1}^n - u_{m-1}^n}{2h} = 0$$

and their computational cells are illustrated in the Figure 23.

So u_m^n is an approximation to $U(x,t)$ at $x = mh, t = mk$.

Assume that U is sufficiently regular and continuous:

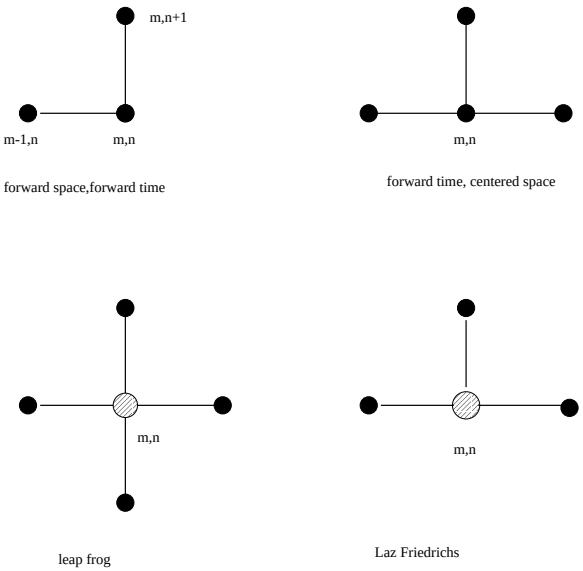

Figure 23: Computational cells for some elementary schemes for the approximation of hyperbolic schemes

take $U(x_m \pm h, t) = U(x_m, t_n) \pm h\dfrac{\partial U}{\partial x}(x_m, t_n) + \dfrac{1}{2}h^2\dfrac{d^2U}{dx^2}(x_m, t_n) + \cdots$

$$\Rightarrow \quad \frac{U(x_m \pm h, t_n) - U(x_m, t_n)}{h} = \frac{\partial U}{dx}(x_m, t_n) + \mathcal{O}(h)$$

$$\Rightarrow \quad \frac{(U_{m+h}, t_n) - U(x_m - h, t_n)}{2h} = \frac{\partial U}{dx}(x_m, t_n)+) + \mathcal{O}(h^2)$$

same procedure leads to finite difference approximation scheme to $\dfrac{\partial}{\partial t}$ derivatives.

Example check convergence of the Lax-Friedrichs scheme:

$$P_{k,h}\phi = \frac{\phi_m^{n+1} - \frac{1}{2}\left(\phi_{m+1}^n + \phi_{m-1}^n\right)}{k} + a\frac{\phi_{m+1}^n - \phi_{m+1}^n}{2h}$$

$$\text{let } \phi_m^n \equiv \phi(t_n, x_m) \text{ then}$$

$$\phi_{m\pm1}^n = \phi_m^n \pm h\phi_x + \frac{1}{2}\phi_{xx} \pm \frac{h^3}{6}\phi_{xxx} + \mathcal{O}(h^4)$$

$$\therefore \quad \frac{1}{2}(\phi_{m+1}^n + \phi_{m-1}^n) = \phi_m^n + \frac{1}{2}h^2\phi_{xx} + \mathcal{O}(h^4)$$

$$\frac{\phi_{m+1}^n - \phi_{m-1}^n}{2h} = \phi_x + \frac{1}{6}h^2\phi_{xxx} + \mathcal{O}(h^4)$$

substituting

$$P_{k,h}\phi = \phi_t + a\phi_x + \frac{1}{2}k\phi_{tt} - \frac{1}{2}\frac{h^2}{k}\phi_{xx} + \frac{1}{6}ah^2\phi_{xxx} + \mathcal{O}\left(h^4 + \frac{h^4}{k} + k^2\right)$$

$$\text{so } P_{k,h}\phi \to 0 \text{ as } k, h \to 0, \text{ i.e. consistent as long as } \frac{h^2}{k} \to 0$$

so what happens when using finite h and k? If k is significantly smaller than h^2 then $\frac{h^2}{k} \geq \mathcal{O}(1)$ quantity. Thus, we would effectively be solving the problem $\phi_t + a\phi_x - L\phi_{xx} = 0$ (due to finite truncation error terms) which is not what we set out to do in the first place!!

A Fundamental Theorem in FD approximations of Hyperbolic PDE's is the **Courant-Friedricks-Lewy Condition (CFL)**, which will be related to the stability of a scheme.

Stability For the homogeneous problem $U_t + aU_x = f$, i.e. with $f = 0$:

definition The IVP for the first order hyperbolic pde $U_t + aU_x = 0$ is well-posed if for any time $T \geq t_0 \quad \exists \quad C_T$ constant such that any solution $U(t, x)$ satisfies

$$\int_{-\infty}^{\infty} |U(t,x)|^2 dx \leq C_T \int_{-\infty}^{\infty} |U(t_0,x)|^2 dx, \text{ for } t_0 \leq t \leq T$$

definition A finite difference scheme $P_{k,h}u_m^n = 0$ for a first-order equation is stable in a stability region $\mathbb{D}$ if there's an integer J such that for any positive

time $T > t_0 \quad \exists \quad C_T$ constant such that

$$h \sum_{m=-\infty}^{\infty} |u_m^n|^2 \leq C_T h \sum_{j=0}^{J} \sum_{m=\infty}^{\infty} |u_m^j|^2$$

$$\text{for } t_0 \leq t_0 + nk \leq T \quad \text{with} \quad (k,h) \in \mathbb{D}$$

$J = 0$ for 1-step schemes and $J > 0$ for multistep schemes, with data at first $J + 1$ levels.

Example

Show that

$$\sum_{m=-\infty}^{\infty} |v_m^{n+1}|^2 \leq (|\alpha| + |\beta|)^2 \sum_{m=-\infty}^{\infty} |v_m^n|^2 \text{ for}$$

$$\text{and thus the scheme } v_m^{n+1} = \alpha v_m^n + \beta v_{m+1}^n$$

$$\text{is stable if } (|\alpha| + |\beta|) \leq 1.$$

□

The Courant-Friedricks-Lewy Condition (CFL)

definition: Explicit FD scheme can be written as

$$v_m^{n+1} = \text{a finite sum of } v_{m'}^{n'} \quad n' \leq n$$

Theorem: For $U_t + aU_x = 0$ with explicit scheme of the form $u_m^{n+1} = \alpha u_{m-1}^n + \beta u_m^n + \alpha u_{m+1}^n$ with $k/h = \lambda$ constant, a necessary and sufficient addition for stability is the CFL condition

$$|a_i \lambda| \leq 1$$

For systems of equations, where u is vector and α, β, α are matrices, we require that $|a_i \lambda| \leq 1$ for all e'values a_i of the matrix a.

Proof: (Heuristic) See Figure 24 Take Scalar Case: Take $|a\lambda| > 1$ first: Consider $(t, x) = (1, 0)$. The solution at $u(1, 0)$ depends on value of $u_0(x)$ at either $x = a$ or $x = -a$ (depending on sign of speed "a"). But from finite difference scheme we have

$$u_0^n \text{depends on } u_m^0 \text{ only for } m \leq n(\text{ by the form of the scheme})$$

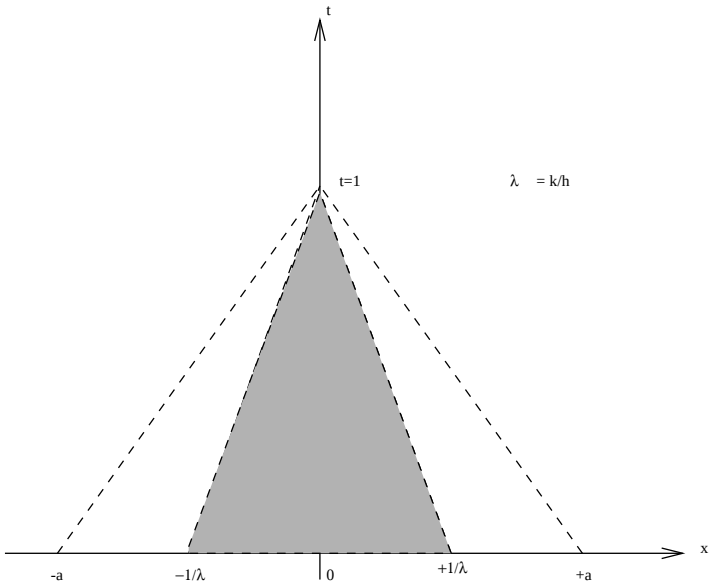

Figure 24: Heuristic proof of theorem, based on the finite time of travel of information from cell to cell.

Since $h = \lambda^{-1}k \Rightarrow mh \leq k\lambda^{-1}n = \lambda^{-1}$ since $k = \dfrac{1}{n}$ (assumes that $t_{\text{final}} = 1$), $\therefore$ u_0^n depends on x only for $|x| \leq \lambda^{-1} < |a|$. Thus u_0^n cannot converge to $u(1,0)$ as $h \to 0$ with $\dfrac{h}{k} = 1$. For $|a\lambda| \leq 1$ things work out (convince yourself).

□

Theorem: (explicit schemes) There's no explicit, unconditionally stable, consistent finite difference schemes for hyperbolic systems of pde's.

Proof: Omitted. □

Remark: Unconditionally stable means that we can choose any k and h and still remain in the region of stability for the particular scheme. In many instances, a physical problem may require that we time-step an approximation over many many time steps. An explicit scheme is attractive here, because it is very efficient in storage (and usually easy to code). However, we need to consider how long a computation is going to actually take in real clock time: if we are restricted by a very small time step, then it may take a very long

time to solve a problem. An alternative is to go to a higher order explicit scheme (but this usually means more communication which is of concern in parallel computing) and this buys us a little longer time steps. However, we might consider a low order implicit scheme which might buy us significantly bigger time steps (but usually a lot more communication). A recent popular alternative are the "Semi-Lagrangian Methods".

Example

$$\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u^{n+1} - u_{m-1}^{n+1}}{h} = 0$$

Implicit Case computational cell

$$(1 + a\lambda)u_m^{n+1} = u_m^n + a\lambda u_{m-1}^{n+1} \quad \lambda = \frac{k}{h}$$

square both sides

$$\begin{aligned}(1 + a\lambda)^2|u_m^{n+1}|^2 &\leq |u_m^n|^2 + 2a\lambda|u_m^n||u_{m-1}^{n+1}| + a^2\lambda^2|u_{m-1}^{n+1}|^2 \\ &\leq (1 + a\lambda)|u_m^n|^2 + a\lambda(1 + a\lambda)|u_{m-1}^{n+1}|^2\end{aligned}$$

Taking sums over all m:

$$(1 + a\lambda)^2 \sum_{m=-\infty}^{\infty} |u_m^{n+1}|^2 \leq (1 + a\lambda) \sum_{m=-\infty}^{\infty} |u_m^n|^2 + a\lambda(1 + a\lambda) \sum_{m=\infty}^{\infty} |u_m^{n+1}|^2$$

$$\therefore \sum_{m=-\infty}^{\infty} |u_m^{n+1}|^2 \leq \sum_{m=\infty}^{\infty} |u_m^n|^2 \quad \therefore \quad \text{stable for all } \lambda \text{ with } a > 0.$$

Analysis of Finite Difference Schemes

Reminder on Fourier Analysis on $\mathbb{R}$

1. Continuous Case

$$\text{Fourier Transform Pair} \begin{cases} \hat{u}(\omega) = \dfrac{1}{\sqrt{2\pi}} \displaystyle\int_{-\infty}^{\infty} e^{-i\omega x} u(x) dx \\ u(x) = \dfrac{1}{\sqrt{2\pi}} \displaystyle\int_{-\infty}^{\infty} e^{i\omega x} \hat{u}(\omega) d\omega \end{cases}$$

2. on a grid of integers $\mathbb{Z}$ or $h\mathbb{Z} = \{hm : m \in \mathbb{Z}\}$

$$\begin{cases} \hat{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-imh\xi} u_m & \text{for } \xi \in [-\pi, \pi] \\ & u(-\pi) = u(\pi), \text{periodic} \\ u_m = \dfrac{1}{\sqrt{2\pi}} \displaystyle\int_{-\pi}^{\pi} e^{im\xi} \hat{u}(\xi) d\xi \end{cases}$$

Sometimes more convenient to express as

$$\hat{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-imh\xi} v_m h \qquad \xi \in [-\pi/h, \pi/h]$$
$$u_m = \frac{1}{\sqrt{2\pi}} \int_{\frac{-\pi}{h}}^{\frac{\pi}{h}} e^{imu\xi} \hat{u}(\xi) d\xi$$

definition $||u||_2 = \sqrt{\displaystyle\int_{-\infty}^{\infty} ||u(x)|^2 dx}$, is the L^2-norm.

Parseval's Theorem states that

$$\int_{-\infty}^{\infty} |u(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{u}(\omega)|^2 d\omega$$

and for the grid functions:

$$||\hat{u}||_h^2 = \int_{-\pi/h}^{\pi/h} |\hat{u}(\xi)|^2 d\xi = \sum_{-\infty}^{\infty} |u_m|^2 h = ||u||_h^2$$

Von-Neumann Analysis of Finite Difference Schemes

Use of Fourier methods in the analysis of finite difference schemes. Fourier methods for the analysis of finite difference schemes are very useful due to their simplicity. They are applicable on all linear problems and somewhat applicable for nonlinear problems. Briefly, the idea is as follows: the finite-dimensional approximation u_m^n on the lattice (mh, nk) of the function $u(mh, nk)$ is decomposed into a superposition of normalized sines and cosines

with wave numbers ξ in the range $\frac{-\pi}{h}$ to $\frac{\pi}{h}$. Thus, each sine/cosine wave is of the form

$$\hat{u}(\xi)e^{ix_m\xi}$$

where $\hat{u}(\xi)$ is the complex amplitude of the ξ^{th} wave. If u depends on both space x_m and on time t_n, then the complex amplitude $\hat{u}(\xi)$ depends on time, hence $\hat{u}^n(\xi)e^{ix_m\xi}$ is the ξ^{th} wave component of u at time $t_n = nk$. Note that the phase $e^{ix_m\xi}$ is of magnitude 1. Hence, for stability all we have to study is each time-dependent complex amplitude $\hat{u}^n(\xi)$.

Take

$$\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_m^n - u_{m-1}^n}{h} = 0$$

$$\text{or } u_m^{n+1} = (1 - a\lambda)u_m^n + a\lambda u_{m-1}^n \quad \lambda = k/h$$

$$\text{take } u_m^n = \frac{1}{\sqrt{2\pi}} \int_{-\pi/h}^{\pi/h} e^{imh\xi}\hat{u}^n(\xi)d\xi$$

and substitute in the above equation:

$$u_m^{n+1} = \frac{1}{\sqrt{2\pi}} \int_{-\pi/L}^{\pi/h} e^{imh\xi} \underbrace{[(1 - a\lambda) + a\lambda e^{-ih\xi}]\hat{u}^n(\xi)d\xi}_{\therefore \quad \hat{u}^{n+1}}$$

$$\left.\begin{aligned} \hat{u}^{n+1}(\xi) &= \underbrace{\left[(1 - a\lambda) + a\lambda e^{-ih\xi}\right]}_{\text{let } g(h\xi)} \hat{u}^n(\xi) \\ &\equiv \underbrace{g(h\xi)}_{\substack{\text{“amplification} \\ \text{factor”}}} \hat{u}^n(\xi) \end{aligned}\right\}$$

amplification factor because its magnitude is the amount that the complex amplitude of each wave in the solution, is amplified in advancing one step in time.

In fact, from the above expression we obtain

$$\hat{u}^n(\xi) = g(h\xi)^n\hat{u}^0(\xi)$$

(137) So returning to $\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_m^n - u_{m-1}^n}{h} = 0$

First, by Perseval's Theorem:

$$h\sum_{m=-a}^{\infty}|u_m^n|^2 = \int_{-\pi/h}^{\pi/h}|\hat{u}^n(\xi)|^2 d\xi$$

$$= \int_{-\pi/h}^{\pi/h}|g(h\xi)|^{2n}|\hat{u}^o(\xi)|^2 d\xi \tag{138}$$

Recall that for first order single step scheme we require that

$$h\sum_{m=-\infty}^{\infty}|u_m^n|^2 \le C_T h\sum_{m=-\infty}^{\infty}|u_m^0|^2 \tag{139}$$

$$\text{for } 0 \le nk \le T \qquad \text{where} \qquad t_n = 0, k, 2k, \ldots$$

for stability.

Comparison of (138) and (139) implies that $|g(h\xi)|^{2n}$ must be suitably bounded. For (137):

$$u_m^{n+1} = u_m^n - a\lambda(u_m^n - u_{m-1}^n)$$

enough to consider

$$\hat{u}_m^n = e^{imh\xi}\hat{u}^n(\xi) \quad \text{(is a wave component of solution)}$$

since linear

$$\hat{u}^{n+1}(\xi)e^{imh\xi} = \hat{u}^n(\xi)e^{imh\xi} - a\lambda\left(\hat{u}_n^n e^{imh\xi} - \hat{u}^m e^{i(m-1)h\xi}\right)$$

$$u^{n+1}(\xi) = \underbrace{[1 - a\lambda(1 - e^{-i\theta})]}_{g(h\xi)}\hat{u}^n(\xi) \qquad \theta = h\xi$$

$$|g(\theta)|^2 = 1 - 4a\lambda(1 - a\lambda)\sin^2\left(\frac{1}{2}\theta\right)$$

if $|g(\theta)|^2 \le 1$ then $0 \le a\lambda \le 1$ $\therefore$

$$h\sum_{m=-\infty}^{\infty}|u_m^n|^2 \le \int_{-\pi/h}^{\pi/h}|u^0(\xi)|^2 d\xi = h\sum_{m=-\infty}^{\infty}|u_m^0|^2$$

□

For this particular example the amplification factor g depended on $\theta = h\xi$ only, but in general it can depend on h and k.

Theorem (Stability, Von Newmann): A one-step constant coefficient scheme is stable if and only if $\exists$ a constant K (independent θ, k, and h) and some positive grid spacings k_0 and h_0 such that

$$|g(\theta, k, h)| \leq 1 + Kk \tag{140}$$

$\forall \theta$, $0 < k < k_0$, $0 < h < h_0$. If g is independent of h, and k then (140) is replaced by

$$|g(\theta)| \leq 1$$

□

Example

$U_t + aU_x - U = 0$ (has solutions that grow with t) using Lax-Friedrichs:

$$\frac{u_m^{n+1} - \frac{1}{2}(u_{m+1}^n + u_{m-1}^n)}{k} + a\frac{u_{m+1}^n - u_{m-1}^n}{2h} - u_m^n = 0$$

$$g(\theta, k, h) = \cos\theta - ia\lambda\cos\theta + k$$

$$|g|^2 = (\cos\theta + k)^2 + a^2\lambda^2\sin^2\theta \leq (1+k)^2 \quad \text{if } |a\lambda| \leq 1$$

so by theorem any stable scheme must have $|g|^2$ larger than 1 for some θ.

Proof: By Parseval's Theorem

$$||u^n||_h^2 = \int_{-\pi/2}^{\pi/h} |g(h\xi, k, h)|^{2n} |u^0(\xi)|^2 d\xi$$

if $|g(h\xi, k, h)| \leq 1 + Kk \therefore$

$$||u^n||_h^2 \leq \int_{-\pi/h}^{\pi/h} (1+Kk)^{2n} |\hat{u}^0(\xi)|^2 d\xi = (1+Kk)^{2n} ||u^0||_h^2$$

Now $n \leq T/k$ so $\quad (1+Kk)^n \leq (1+Kk)^{T/k} \leq e^{kT}$

$\therefore ||u^n||_n \leq e^{Kt} ||u^0||_h$ which is the condition

$$h \sum_{m=-\infty}^{\infty} |u_m^n|^2 \leq C_T h \sum_{j=0}^{J} \sum_{m=\infty}^{\infty} |u_m^j|^2 \quad \text{for } 0 \leq nk \leq T \tag{141}$$

Now we prove that if inequality (140) cannot be satisfied for any value of $K \Rightarrow$ scheme is not stable. To do so, we can show that any amount of growth in the solution, that is, we show that the stability inequality (141) cannot hold.

If for some positive value C there's an interval of θ's, $\theta \in [\theta_1, \theta_0]$ and $h \in (0, h_0]$ and $k \in (0, k_0]$ with $|g(\theta, k, h)| \geq 1 + Ck$ then we construct a function v_m^0 as

$$\hat{u}^0(\xi) = \begin{cases} 0 & \text{if } h\xi \notin [\theta_1, \theta_2] \\ \sqrt{h(\theta_2 - \theta_1)^{-1}} & \text{if } h\xi \in [\theta_1, \theta_2] \end{cases}$$

then

$$\begin{aligned} ||u^n||_h^2 &= \int_{-\pi/h}^{\pi/L} |g(h\xi, k, h)|^{2n} |\hat{u}^0(\xi)|^2 d\xi \\ &= \int_{\theta_1/h}^{\theta_2/h} |g(h\xi, k, h)|^{2n} \frac{h}{\theta_2 - \theta_1} d\xi \geq (1 + Ck)^{2n} \\ &\geq \frac{1}{2} e^{2TC} ||u^0||_h^2 \text{ for } n \text{ near } T/k. \end{aligned}$$

This shows that the scheme to be unstable if C can be arbitrarily large.

□

Corollary: If a scheme as in previous theorem is modified so that the modifications result only in the addition to the amplification factor of terms that are $\mathcal{O}(k)$ uniformly in ξ, then the modified scheme is stable if and only if the original scheme is stable.

Proof: If g is the amplification factor for the scheme and satisifies $|g| \leq 1 + Kk$, then the amplification factor or the modified scheme g' satisfies

$$|g'| = |g + \mathcal{O}(k)| \leq 1 + Kk + Ck = 1 + K'k \tag{142}$$

Hence the modified scheme is stable if the original scheme is stable and vice versa. □

Stability for variable coefficients

Take $U_t + a(x,t)U_x = 0$ as an example.

The general procedure is to consider the problem with a as a frozen coefficient for each x, t values in question. If each frozen coefficient case is stable then the scheme is stable. For example, the CFL condition would require

$$|a(t_n, x_m)|\lambda \leq 1$$

for all t_n, x_m in computational lattice.

Remark: Numerical vs Dynamic Stability $\rightarrow$ Numerical stability refers to the behavior of approximations to a grid projected equation over a finite time interval as the lattice is refined. Dynamic stability refers to the behavior of solutions of PDE as $t \rightarrow \infty$.

Example

$$u_t + au_x + bu = 0 \quad x \in \mathbb{R}^1 \\ , t > 0$$

for $b > 0$, solution is dynamically stable (bounded) since solution decays as $t \rightarrow \infty$. (Show this using Fourier methods).

for $b > 0$, solution is unstable.

For above equation a stable numerical scheme would be one in which the approximation converges to exact solution for any b as $k, h \rightarrow 0$.

□

Comments on Instabilities in Hyperbolic Equation Approximation

1) As always, non-convergent schemes are useless.

2) Take Lax-Friedrichs: $|g(\theta)|^2 = \cos^2\theta + a^2\lambda^2 \sin^3\theta$. The maximum value of g is attained when $\theta = \dfrac{\pi}{2}$, where $|g| = 1.6 \Rightarrow$ so instabilities are related to high frequency oscillations.

3) In general (not a theorem) $\Rightarrow$ instabilities will manifest themselves as rapid growth of high wave numbers and would then be more evident if the initial data contains high amplitude high wave number modes (e.g. non-smooth data). Also, the instabilities will have wavelengths that are comparable or commensurate to the grid space in x. Also, the instability phenomenom will be local in nature but propagate in time.

Of course, it will eventually swamp the solution, but for close times after the onset of the instability, they are local.

4) Having a good feel for this allows one to discern between programming errors and improper schemes.

5) The other considerations related to stability are that schemes with very restrictive conditions on step sizes, in particular, on time steps, will require more computational effort in calculation. This is ok, but aside from stability, we also need to consider the dissipation and dispersion in the scheme.

6) Stability of a scheme requires that this issue be checked carefully at the boundaries. One of the most common sources of instabilities comes from using inappropriate boundary conditions.

Example

Take Lax-Friedrichs on

$$U_t + aU_{xxx} = f.$$

The finite difference approximation is

$$u_m^{n+1} = \frac{1}{2}(u_{m+1}^n + u_{m-1}^n) - \frac{1}{2}akh^{-3}(u_{m+2}^n - 2u_{m+1}^n + 2u_{m-1}^n - u_{m-2}^n) + kf_m^n$$

this scheme is consistent if $h^2/k \to 0$ as h and k tend to 0.

The amplification factor in this case is

$$g(\theta) = \cos\theta + i4akh^{-3}\sin\theta\sin^2\frac{1}{2}\theta$$

so scheme is stable if $\dfrac{4|a|k}{h^3}$ is bounded.

However the consistency condition $\frac{h^2}{k} \to 0$ as h and $k \to 0$ and stability condition $\dfrac{4|a|k}{h^3}$ bounded cannot be both satisfied.

$\therefore$ Scheme is not convergent. □

Truncation Error and Order of Accuracy for FD Schemes

definition: A scheme $P_{k,h}u = R_{k,h}f$ that is consistent with $PU = f$ is accurate of order p in time and q in space if for any smooth $\phi(t,x)$

$$P_{k,h}\phi - R_{k,h}P\psi = \prime(k^p) + \mathcal{O}(h^q)$$

We say scheme is order (p,q). □

Example Crank-Nicholson

Take

$$U_t = \frac{U(t+k,x) - U(t,x)}{k} + \mathcal{O}(k^2)$$

for

$$U_t(t + \frac{1}{2}k, x)$$

Take

$$\begin{aligned} U_x = (t + \frac{1}{2}k, x) &= \frac{U_x(t+\frac{1}{2}k,x) + U_x(t,x)}{2} + \mathcal{O}k^2 \\ &= \frac{1}{2}\left[\frac{U(t+k,x+h) - U(t+k,x-h)}{2h} + \frac{U(t,x+h) - U(t,x-h)}{2h}\right] \\ &+ \mathcal{O}(k^2) + \mathcal{O}(h^2) \end{aligned}$$

Using these approximate $U_t + aU_x = f$ about $\left(t + \frac{1}{2}k, x\right)$ we get

$$\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_{m+1}^{n+1} - u_{m-1}^{n+1} + u_{m+1}^n - u_{m-1}^n}{4h} = \frac{f_m^{n+1} + f_m^n}{2}$$

And is an order $(2,2)$ scheme.

Exercise

For $U_t + aU_x = 0$ show that Crank-Nicholson has an amplification factor

$$g(\theta) = \frac{1 - i\frac{1}{2}a\lambda\sin\theta}{1 + i\frac{1}{2}a\lambda\sin\theta} \text{ and is unconditionally stable.}$$

Exercise

Show that Lax-Wendroff is consitent with $U_t + aU_x = f$:

$$g(\theta) = 1 - 2a^2\lambda^2\sin^2\frac{1}{2}\theta - ia\lambda\sin\theta$$

stable if $|a\lambda| \leq 1$ and of order $(1,2)$

$$\text{Lax Wendroff: } \frac{u_m^{n+1}-u_m^n}{k}+a\frac{u_{m+1}^n-u_{m-1}^n}{2h}-\frac{a^2k}{2h^2}(u_{m+1}^n-2u_m^n+u_{m-1}^n)$$
$$=\frac{1}{2}\left(f_m^{n+1}+f_m^n\right)-\frac{ak}{4h}\left(f_{m+1}^n-f_{m-1}^n\right)$$

□

Order of Accuracy: The choice of norm is problem-dependent. Could use our grid L_2 norm. Then

Error$(t) = ||U(t,\cdot)-u^n||_h = (h\sum_m |U(t,x_m)-u_m^n|^2)^{\frac{1}{2}} = 0(h^r)$ gives the accuracy of the "solution" u^n as an approximation to exact solution $U(t,x)$ on the grid. The usefulness of the above norm is that we should get the order of accuracy to be equal to order of truncation of scheme FOR SMOOTH DATA.

Boundary Conditions

Numerical schemes may require points outside of computational domain. This happens at boundary conditions. Suppose we are solving a problem over a space grid indexed by $m = 0, 1, \cdots M$. Let u_m^n be an approximation to $U(x_m, t_n)$. Hence the edge variables are u_0^n and u_M^n. For example, suppose we have a scheme that requires u_{M+1}^n in order to determine u_m^n. This can happen when $m = M$, i.e. at the edge of the domain and we will have that u_m^n is determined by u_{M+1}^n as well as by interior grid quantities.

Numerical boundary conditions should be some form of extrapolation that determines the solution on the boundary in terms of the solution in the interior. For example:

$$\text{some numerical b.c.'s} \begin{cases} u_M^{n+1}=u_0^{n+1} & \text{Periodic} \\ u_M^{n+1}=u_{M-1}^{n+1} & \text{simple extrapolation} \\ u_M^{n+1}=2u_{M-1}^{n+1}-u_{M-2}^{n+1} & \\ u_M^{n+1}=u_M^n-a\lambda(u_M^n-u_{M-1}^n) & \text{(quasi-characteristics)} \end{cases}$$

We will consider boundary conditions further when we discuss Parabolic equations (see 0.5.1).

There are three aspects to numerical boundary conditions

- The real boundary conditions, along with initial data and the pde, should lead to a well-posed problem.
- Need to come up with a numerical approximation of the physical boundary conditions, and this may lead to approximate boundary conditions which have their own inherent truncation error as well as round-off errors.
- A stable numerical scheme may become unstable by a bad choice of boundary data or a bad choice of approximating scheme for the boundary data!!

Recall that von Neumann stability only gives stability of IVP, so we need to consider stability of the scheme in the neighborhood of the boundaries: **always work out the stability issues on each boundary. First do these separately, and then check that they all work together**. When we do parabolic problems we will use elementary matrix methods to infer whether or not a particular choice of b.c. will lead to instabilities when coupled to a particular scheme.

But now we revert to two important aspects of the quality of an approximating scheme, which are most important in hyperbolic problems but that also are considerations in other types of evolutionary problems. These are: Dissipation and Dispersion.

Dissipation

Roughly speaking, numerically-induced energy loss. If it is worse that the dissipation inherent in the PDE, it will lead to incompatible approximations to the real solution. The requirement that a scheme have $|g(h\xi)| \leq 1$ ensures stability. If $|g(h\xi)| = 1$ for all $h\xi$, it means that the amplitude of the mode ξ is not affected by time-stepping: it does not get diminished. However, if $|h(\xi)| < 1$ for some or all $h\xi$, a slight or large loss in amplitude is incurred. This is dissipatioin (numerical) and is an artifice of the scheme rather than of the equation being approximated.

For evolutionary PDE's, as can be imagined, if a scheme has more numerical dissipation than is inherent in the PDE being approximated, the solution will

eventually be different in amplitude and phase to the approximation solution provided by a scheme with less dissipation. The phase phenomenom must be considered too, since the scheme will in general have different rates of dissipation for different modes.

In order to measure to dissipation concretely, we have to agree on a definition of dissipation. This is one possibility:

definition: Let $\theta = \xi h$. A scheme is dissipative of order $2r$ if there exists a positive contant c, independent of h and k, such that each amplification factor $g_\nu(\theta)$ satisfies

$$|g(\theta)|^2 \leq 1 - c \sin^{2r}\left(\frac{1}{2}\theta\right)$$

Other definitions will replace $\sin^{2r}\left(\frac{1}{2}\theta\right)$ by $|\theta|^{2r}$.

Example

This happens at boundary conditions. Suppose we are solving a problem over a space grid indexed by $m = 0, 1 \cdots M$. Let u_m^n be an approximation to $U(x_m, t_n)$. Hence the edge variables are u_0^n and u_M^n.

Lax-Wendroff:

$$|g(\theta)|^2 = 1 - 4a^{e2}\lambda^2(1 - a^2\lambda^2)\sin^4\frac{1}{2}\theta$$

For $|a\lambda| = 1$ we have $|g(\theta)| = 1$, nondissipative. But for $0 < |a\lambda| < 1$ the scheme is of order 4 in dissipation.

Example: Show that

Leapfrog and Crank Nicholson $\rightarrow$ both non-dissipative since their amplificaiton factors are identically 1 in magnitude.

Example

Lax-Friedrichs: show that $|g(\theta)| = 1$ for $\theta = 0$ and π, but less than 1 for their values $\Rightarrow$ dissipative.

Remark: Sometimes dissipation is good. It may also be added to schemes in order to stabilize them. For example, adding

Dissipation may also be added to schemes in order to stabilize them. For

example, adding

$$\frac{\varepsilon}{2k}\left(\frac{1}{2}h\delta\right)^4 v_m^{n-1}$$

where $\delta^2 v_m^{n-1} \equiv v_{m+1} - 2v_m + v_{m=1}$

and $\varepsilon \ll 1$ leads to

$$g_\pm = -a\lambda\sin\theta \pm \sqrt{1 - a^2\lambda^2\sin^2\theta - \varepsilon\sin^4\frac{1}{2}\theta}$$

if $\varepsilon < 1 - a^2\lambda^2$ scheme is stable and of $\mathcal{O}(4)$ in dissipation.

Dispersion There are PDE's that have dispersive terms (KdV, Nonlinear Schrodinger Equation, etc). In hyperbolic problems, these dispersive terms force each Fourier mode to travel at different speeds. Hence, if a wave that was compact at some time is subjected to dispersion (and is not balanced by other other effects, such as could be possible with nonlinearity or dissipation), will eventually spread out in space and time. An example of a real physical system in which dispersive effects are readily observed is: throw a rock into a pond and watch the concentric waves propagate out of the center of impact. Far from the center you see that waves of different wavelengths will separate. This would not happen if the surface of the lake, which is capable of supporting waves, were not dispersive. In the absence of dispersion the initial disturbance set up by the rock would propagate out as a single and compact ring of waves.

Dispersion can also be caused unwittingly by certain numerical approximations to equations. If it is unwanted, it is a form of distortion and it turns out a fairly important one. Suppose we were solving the one-way wave equation with constant speed. In Figure 25, which would correspond to the approximation with a numerical method with zero dispersion, we have the initial data, which can be thought of as a superposition of waves (via Fourier methods) of wavenumber κ and corresponding frequency ω all traveling at speed c, constant. Hence, the dispersion relation $\omega = \kappa c$, where c is constant. In Figure 26 we would have the same initial data with each wave of component traveling at speed $c(\kappa)$ and the initial data would then spread and distort.

c is fixed constant speed. k is the wave number and ω the frequency.

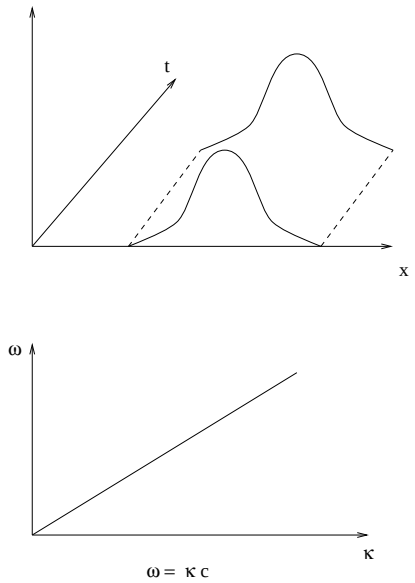

Figure 25: Non-dispersive approximation to the one-way wave equation with wave speed c, constant. The dispersion relation is a straight line with slope c

Take

$$U(t,x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega x} e^{-i\omega a t} \hat{U}_0(\omega) d\omega \tag{143}$$

This is a solution to

$$\begin{aligned} U_t + aU_x &= 0 \\ U(x,0) &= U_0(x) \end{aligned}$$

Here a is constant.

From (143) we conclude that

$$\hat{U}(t+k,\omega) = e^{-i\omega k} \widehat{U}(t,\omega). \tag{144}$$

A one-step finite difference scheme gives

$$\hat{U}^{n+1} = g(h\xi)\hat{U}^n \tag{145}$$

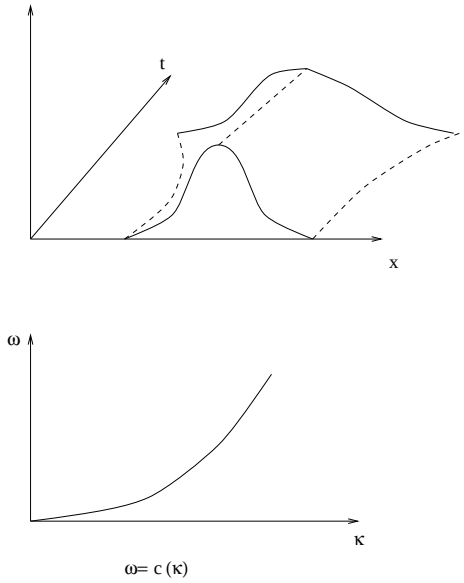

Figure 26: Dispersive approximation to the one-way wave equation with wave speed c no longer a constant, hence the dispersion relation is $\omega = c(\kappa)$. The local slope of this function below is the speed, which shows that the higher frequencies are traveling faster than the slower ones.

Comparing (144) and (145) we expect $g(h\xi)$ to be a good approximation to $e^{-1\xi ak}$. Let's write $g(h\xi)$ in term s of magnitude and phase:

$$g(h\xi) = |g(h\xi)|e^{-i\xi\alpha(h\xi)k}$$

"phase speed" is the speed at which waves of wave number ξ are propagated by the FD scheme.

If $\alpha(h\xi) = a$, constant, for all $\xi \Rightarrow$ waves would propagate at correct speed ... thus is what we expect from analytical solution of $u_t + au_x = 0$.

The "phase error" could be a measured by $a - \alpha(h\xi)$.

When the waves travel at different speeds, we say the FD scheme is dispersive and can use the phase error to assess how badly this affects the solution. When the waves travel at constant speed $\alpha_0 = k'a$, where k' is a constant and a is the wave speed in $u_t + au_x = 0$, we say the FD scheme is non-dispersive.

Example

Lax-Wendroff: $g = 1 - 2(a\lambda)^2 \sin^2 \frac{1}{2}h\xi ia\lambda \sin h\xi$

$$\text{and so take } [\alpha(h\xi)\xi k] = \frac{a\lambda \sin h\xi}{1 - 2(a\lambda)^2 \sin^2 \frac{1}{2}h\xi}$$

But we want $\dfrac{\tan^{-1}}{\xi k}\left[\dfrac{a\lambda \sin h\xi}{1 - 2(a\lambda)^2 \sin^2 \frac{1}{2}h\xi}\right] = \alpha(h\xi)$

Take the low wave number limit $\xi \to 0$

$$\alpha(h\xi) = a\left\{1 - \frac{1}{6}(h\xi)^2[1 - (a\lambda)^2] + 0(h\xi)^4\right\}$$

So for $h\xi$ small and $(a\lambda) < 1, \alpha(h\xi) < a$. Also if $|a\lambda| \to 1$ then the dispersion is smaller.

Example Find dispersion behavior for larger $\xi \to h^{-1}\pi$.

Remarks

(1) In general, for hyperbolic problems, we usually want to take $|a\lambda|$ close to stability limit: usually gives largest time steps and in general, for

dissipative schemes, the least dissipation. But more importantly, most likely would yield the smallest dissipation and disperse errors.

(2) The comments here apply broadly to all evolutionary PDE approximations.

(3) The leap-frog is an example of a scheme with no dispersion error **when** $a\lambda = 1$. It doesn't have dissipation either when $a\lambda = 1$. However, the leap-frog method have a couple of notorious problems: a) it can have bad stability problems when coupled to certain boundary conditions. b) being that it is a 2nd order-in-line method, its approximation is made of 2 traveling wave solutions, traveling in opposite directions, generally. However, if the hyperbolic problem being approximated only admits 1-way wave solutions, a single wave, care must be exercised in either not exciting the spurious solution or in actively suppressing it ... one popular suppressing technique is to revert to a single Euler every 100's or 1000's time steps and then reverting back to Leapfrog (doing too many Euler will defeat the purpose of using leapfrog and will generate a lot of dissipation). There are other techniques for this, such as using time- and/or space-averaging filters.

Group Velocity and Propagation of Wave Packets

The group velocity is the speed at which the energy in a wave packet travels at. It is a useful concept in nonlinear and dispersive equations. It can be used to explain some rather striking behavior of certain schemes, including the explanation of certain instabilities caused by boundary conditions (see Trefethen).

We've seen that dispersive FD schemes will cause a pure wave with wave number ξ_0 to travel with phase speed $\alpha(h\xi_0)$. We want to know what is the velocity of propagation of the center of mass of a wave packet.

The scheme group velocity is

$$\gamma(\theta) \equiv \frac{d}{d\theta}(\theta\alpha(\theta)), \text{where} \quad \theta = h\xi$$

A wave packet example suppose initial data of the form

$$u(0, x) = e^{i\xi_0 x} p(x)$$

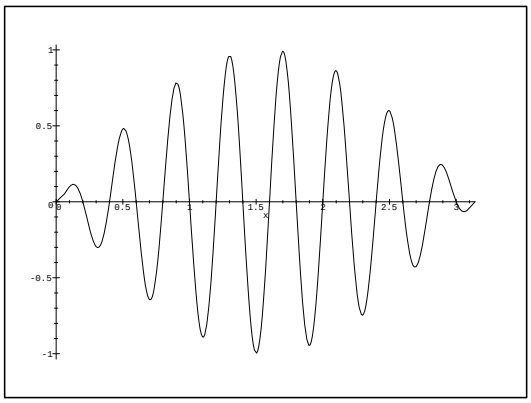

Figure 27: Wave Packet

where $p(x)$ is a relatively smooth function decaying rapidly about its centor of mass. For

$$u_t + au_x = 0$$

the solution is $u(t,x) = e^{i\xi_0(x-at)}p(x-at)$

$p(x)$ is the envelope of the wave packet and $e^{i\xi_0 x}$ is the carrier wave, see Figure 27

For a finite difference approximation below, γ is the group velocity:

$$u(t,x) = e^{i\xi_0(x-\alpha(h\xi_0)t)}p(x-\gamma(h\xi_0)t)$$

(in the case of zero dissipation). Note that since $\alpha(h\xi) \to a$ as $h \to 0$ we have that $\gamma(h\xi_0) \to a$ as $h \to 0$ and $u \to U$, the exact solution.

Conservation Laws

(146)

Is a system of equations of the form $\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0} & \mathbf{u} \in \mathbb{R}^d \\ \mathbf{u}(\mathbf{0}, \mathbf{x}) = \mathbf{u_0}(\mathbf{x}) & \mathbf{x} \in \mathbb{R}^n, t > 0 \end{cases}$

It's a conservation law since the flux $\mathbf{F}$ will be constant if $d\mathbf{u}/dt = 0$. Conservation law equations are some of the most important and ubiquitous equations of Physics.

Conservation Laws have a connection to the wave equation. Take the case of one space dimension: if $\mathbf{f}$ is smooth enough, with $x \in \mathbb{R}^1$ for example:

$$\frac{\partial \mathbf{u}}{\partial t} + [f(\mathbf{u})]_{\mathbf{x}} = \mathbf{0} \Rightarrow \frac{\partial \mathbf{u}}{\partial \mathbf{t}} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \mathbf{0}$$

$$\frac{\partial f}{\partial \mathbf{u}} = \mathbf{c} \text{ is a "speed"}$$

then $\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{c}\dfrac{\partial \mathbf{u}}{\partial x} = 0$, the 1-way wave equation.

A very good source of information on approximating solutions numerically to (146) is LeVeque's book.

We'll do a classic example: Burger's Equation in 1-D

Take

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}(u^2) = 0 \Rightarrow \frac{\partial u}{\partial t} + u u_x = 0$$

with

$$u(x,0) = g(x) \qquad x \in \mathbb{R}^1$$

and assume that $\displaystyle\int_{-\infty}^{\infty} |g(x)|^2 dx < \infty$ and $g(x)$ is differentiable.

Now can use analysis at beginning of this section to show that the characteristics corresponding to $g(x) =$ a step function will look like those in Figure 28 and at the step rise we get a rarefaction. Can you come up with a $g(x)$ that would lead to a shock, i.e. a crossing of characteristics?

There are many techniques for approximating solutions to this and other conservation laws $\Rightarrow$ they are front-tracking techniques shape-preserving, flux-limited, etc. But most are based on solving the Riemann problem locally, i.e., following the characteristics locally.

One such family of methods: GODUNOV-METHODS, which is a mildly dissipative method.

Take Δt_n be the step size that goes from n to $n+1$ time level. The time step size is variable.

let $u_m^0 = \dfrac{1}{\Delta x}\displaystyle\int_{(m-\frac{1}{2})\Delta x}^{(m+\frac{1}{2})\Delta x} g(x)dx$

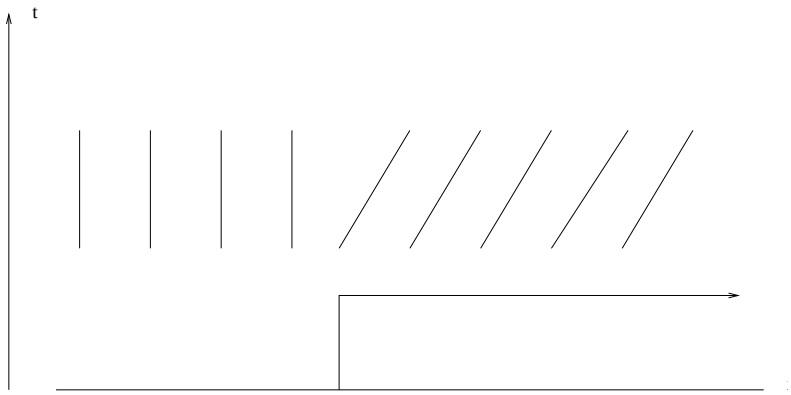

Figure 28: $g(x) = \theta(x)$, a step, leading to a rarefaction.

Suppose all u_m^n are known, we construct a piecewise constant function $w^{[n]}(\cdot, t_n)$ by letting it equal u_m^n in each interval.

Let $\Pi_m \equiv \left(x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}}\right]$ and evaluate the exact solution to the Riemann problem ahead of $t = t_n$.

The idea is to let each interval Π_m "propagate" in the direction determined by its characteristics.

Choose a point $(x, t), t \geq t_n$. There are 3 possibilities

(1) $\exists\, !m$ such that the point is reached by a characteristic emanating from Π_m. Since characteristics propagate constant values, the solution of the Riemann problem at this is u_m^n.

(2) $\exists\, ! \, m$ such that the point is reached by characteristics emanating from the intervals Π_m and Π_{m+1}. In this case, as the 2 intervals "propagate" in time, they are separated by a shock. The shock advances along a straight line starting at $\left(m + \dfrac{1}{2}\right)\Delta x$ and whose slope is the average slopes in Π_m and Π_{m+1}, i.e. $\dfrac{1}{2}\left(u_m^n + u_{m+1}^n\right]$.

Let this line be ρ_m. The value at (x, t) is u_m^n if $x < \rho_m(t)$ and u_{m+1}^n if $x > \rho_m(t)$.

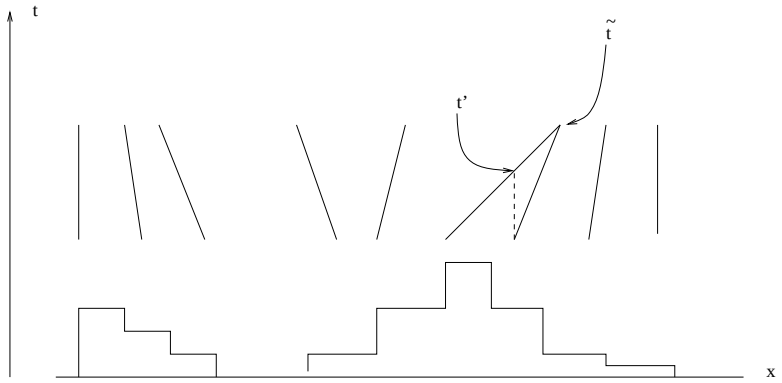

Figure 29: Piece-wise constant approximation of $g(x)$ and the characteristics ρ's emanating from such a $g(x)$, pictured above $g(x)$.

(3) Characteristics from more than 2 intervals reach the point (x,t). In this case, cannot assign a value to the point.

Simple geometry demonstrates that (3) which we must avoid occurs for $t > \tilde{t} > t_n$ is the lowest solution to equation $\rho_m(t) = \rho_{m+1}(t)$ for some m.

This can be seen in Figure 29

let t' be the time of an encounter between the first encounter. Choose

$$t_{n+1} \in (t_n, t') \text{and } \Delta t_n = t_{n+1} - t_n$$

Since $t_{n+1} \in (t_n, t'\,\big]$ cases (143)and (144) can be used to construct a solution $w^{[n]}(x,t) \forall t_n \leq t \leq t_{n+1}$. Choose the

$$u_m^{n+1} = \frac{1}{\Delta x} \int_{(m-\frac{1}{2})\Delta x}^{(m+\frac{1}{2})\Delta x} w^{[n]}(x, t_{n+1}) dx \tag{147}$$

This integral can be calculated.

Disregarding shocks $w^{[n]}$ obeys Burger's Equation for $t \in [t_n, t_{n+1}] \therefore$

$$\frac{\partial w^{[n]}}{\partial t} + \frac{1}{2}\frac{\partial (w^{[n]})^2}{\partial x} = 0 \Rightarrow w^{[n]}(x, t_{n+1}) = w^{[n]}(x, t_n) - \frac{1}{2}\int_{t_n}^{t_{n+1}} \frac{\partial}{dx}[w^{[n]}]^2 dt$$

substituting (147) results in

$$u_m^{n+1} = \frac{1}{\Delta x} \int_{m+\frac{1}{2}\Delta x}^{(m+\frac{1}{2})\Delta x} \left\{ w^{[n]}(x, t_n) - \frac{1}{2}\int_{t_n}^{t_{n+1}} \frac{\partial}{\partial x}[w^{[n]}]^2 dt \right\} dx$$

Since Du^n_{n+m} has been obtained by an averaging procedure given in we have after exchanging the order of integration

$$\begin{aligned} u^{n+1}_m &= u^n_m - \frac{1}{2\Delta x}\int_{t_n}^{t_{n+1}} \int_{(m-\frac{1}{2})\Delta x}^{(m+\frac{1}{2})\Delta x} \frac{\partial}{\partial x}[w^{[n]}]^2 dx dt \\ &= u^n_m - \frac{1}{2\Delta x}\int_{t_n}^{t_{n+1}} \Big\{ \Big[w^{[n]}\Big(\Big(m+\frac{1}{2}\Big)\Delta x, t\Big)\Big]^2 - [w^{[n]}\Big(m-\frac{1}{2}\Big)\Delta x, t\Big)\Big]^2 \Big\} dt \end{aligned}$$

Recall our definition of t_{n+1}. No vertical line segments $\left((m+\frac{1}{2})\Delta x, t\right), t \in [t_n, t_{n+1}]$, may cross the discontinuties ρ_j $\therefore$ the value of $w^{[n]}$ across each such segment is constant – equaling u^n_m or u^n_{m+1} (depending on the slope of ρ_m: if it points rightwards it is u^n_m. Otherwise u^n_{m+1}

Denote this value by $\chi_{m+\frac{1}{2}}$ then

$$u^{n+1}_m = u^n_m - \frac{1}{2}\frac{\Delta t_n}{\Delta x}\left(\chi^2_{m+\frac{1}{2}} - \chi^2_{m-\frac{1}{2}}\right)$$

This is the simplest, first order Godunov scheme. □

In general methods for the approximation of conservation laws should follow characteristics...solve the Riemann problem locally. Godunov is about the local determination of the upwind direction. Another popular upwinding technique is the ENO switch: (see Osher and Engquist) (which is a member of a family of nonlinear techniques that are known as Total Variation Diminishing (TVD) schemes and are very effective in modeling shocks since they have little or no ringing at discontinuities:

$$f_{-(y)} \equiv [\min(y,0)]^2 \quad f_{+(y)} \equiv [\max(y,0)]^2 \qquad y \in \mathbb{R}$$

to get $\dfrac{\partial u}{\partial t} + \dfrac{1}{\Delta x}\big[\Delta_+ f_-(u_m) + \Delta_- f_+(u_m)\big] = 0$

if $u_{m-1}, u_m, u_{m+1} > 0 \rightarrow$ characteristic propagate right $\Rightarrow$

$$\Delta_+ f_-(u_m) = 0 \text{ and } \Delta_- f_+(u_m) = [u_m]^2 - [u_{m-1}]^2$$

if $u_{m-1}, u_m, u_{m+1} < 0 \rightarrow$ characteristic propagate left $\Rightarrow$

$$\Delta_+ f_-(u_m) = [u_{m+1}]^2 - [u_m]^2 \quad \text{and } \Delta_- f_+(u_m) = 0.$$

Again, scheme determines upwind direction locally.

0.5 PARABOLIC EQUATIONS AND THE ADVECTION-DIFFUSION EQUATION

The simplest example is the "Heat Equation"
Let $U = U(x,t)$, and $t > 0$. The Heat Equation is

$$\begin{cases} U_t = bU_{xx} & b > 0 \quad \text{real, called "dissipation constant"} \\ U(0,x) + U_0(x) \end{cases}$$

plus boundary values in x. It is a boundary-initial value problem, but for now, take $x \in \mathbb{R}^1$.

0.5.1 Properties of the Solution

Expect solutions to get smoother as $t \to \infty$. To see this take Fourier transform, with $\widehat{u}(\omega,t) = \mathcal{F}(u(x,t))$ then

$$\widehat{U}_t = -b\omega^2 \widehat{U}$$

integrating and using initial data:

$$\widehat{U}(t,\omega) = e^{-bw^2 t}\widehat{U}_0(\omega)$$

$$\therefore \quad U(t,x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega x} e^{-b\omega^2 t} \widehat{U}_0(\omega) d\omega. \tag{148}$$

Using $\widehat{U}_0(\omega) = \displaystyle\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-i\omega y} U_0(y) dy$ in (148), interchanging the intergrations, one can show that (148) is equal to

$$U(t,x) = \frac{1}{\sqrt{4\pi bt}} \int_{-\infty}^{\infty} e^{-(x-y)^2/4bt} U_0(y) dy.$$

Hence, solution broadens and dissipates at a rate of $\alpha = 1/\sqrt{t}$ this estimate corresponds to 1 space diversion. Can you estimate the rate of dissipation in time in 2 and 3 space dimensions?

An important equation related to both hyperbolic and parabolic equations is the advection-diffusion equation

$$U_t + aU_x = bU_{xx} \tag{149}$$

To solve, let $y = x - at$ and set

$$w(t, y) = U(t, y + at)$$

then $w_t = U_t + aU_x = bU_{xx}$

and $w_y = U_x \quad w_{yy} = U_{xx}$

$$\text{so } w_t = bw_{yy} \tag{150}$$

Hence, since $U(t, x) = w(t, x - at)$ the solution of (149) when examined in a moving coordinate system moving with speed a, is (150). Hence the solution travels with speed a and diffuses with strength b.

General (Petrovskii-form) Parabolic Equation

$$U_t = BU_{xx} + \Delta U_x + CU + F(t, x) \tag{151}$$

B has eigenvalues with all positive real parts and Δ is the Laplacian operator.

For (151), the following estimate holds:

$$\int_{-\infty}^{\infty} |U(t,x)|^2 dx + \int_0^t \int_{-\infty}^{\infty} |U_x(s,x)|^2 dx ds$$
$$\leq C_T \Big(\int_{-\infty}^{\infty} |U(0,x)|^2 dx + \int_0^t \int_{-\infty}^{\infty} |F(b,x)|^2 dx ds \Big)$$

for some $0 \leq t \leq T$. C_T is a constant which may depend on T.

Boundary Conditions for Parabolic equations

$T_0 U = b_0$ Dirichlet-type

$T_1 \dfrac{dU}{dx} + T_2 U = b_1$ Robin-type

Here, T_0 is $d_0 \times d$ matrix and

T_1 and T_2 are $(d - d_0) \times d$ matrices

0.5.2 Finite Difference Schemes

Take $U_t = bU_{xx} + f(x,t)$. Let $u_m^n = U(nk, mh) = U(t_k, x_m)$

Let δ_x be the difference operator, such that

$$\delta_x u_m^n = u_{m+\frac{1}{2}}^n - u_{m-\frac{1}{2}}^n$$

similarly, $\delta_t u_m^n = u_m^{n+\frac{1}{2}} - u_m^{n-\frac{1}{2}}$.

Exercise: Show that $\delta_x^2 u_m^n = u_{m+1}^n - 2u_m^n + u_{m-1}^n$

Backward-time/central space approximation:

$$\frac{u_m^{n+1} - u_m^n}{k} = \frac{b}{h^2}\,\delta_x^2 u_m^{n+1} + f_m^n$$

Exercise

Show that the above scheme is unconditionally stable of order $(1,2)$, and dissipative when $\mu \equiv \dfrac{k}{h^2}$ is bounded away from 0.

Crank-Nicholson (CN)

An old-time favorite:

$$\frac{V_m^{n+1} - u_m^n}{k} = \frac{1}{2h^2} b\left(\delta_x^2 u_m^{n+1} + \delta_x^2 u_m^n\right) + \frac{1}{2}\left(f_m^{n+1} + f_m^n\right)$$

Exercise

Show that CN is implicit, unconditionally stable, of order $(2,2)$. Furthermore, show that it is dissipative of order 2 if μ is constant, but not dissipative if $\lambda = \dfrac{k}{h}$ constant.

Boundary Conditions

Can have tremendous effect on solution. Choice dictated by physics and well-posedness.

Numerical approximations to boundary condition's must be done with great care since they are potentially capable of making an otherwise stable scheme unstable.

There's usually no difficulty implementing Dirichlet type boundary conditions: take boundary conditions at $x = 0$, for simplicity:

$$U(t,0) = b(t) \text{ approximated as } u_0^n = b^n$$

No problem with periodic boundary condition's: assume other boundary at $x = L > 0$:

$$U(t,0) = U(t,L) \text{ approximated as } u_0^n = u_{m+1}^n$$

if there are $M+1$ gridpoints.

What about Neumann? Take $x = 0$ and $x = 1$ as boundaries:

Consider first $\frac{\partial U}{dx}|_{x=0} = a_0$ and $\frac{\partial U}{\partial x}|_{x+1} = a_1$ A first-order approximation would be $\frac{u_1^n - u_0^n}{h} = a_0$ and $\frac{u_M^n - U_{m-1}^n}{h} = a_1$.

Using ghost values u_{-1}^n and u_{M+1}^n, a second-order approximation

$$\frac{u_1^n - u_{-1}^n}{2h} = a_0 \qquad \text{and} \qquad \frac{u_{M+1}^n - u_{M-1}}{2h} = a_1$$

One can also use a 2^{nd}-order approximation, the 1-sided

$$\frac{-3u_0^n + 4u_1^n - u_2^n}{2h} = a_0$$

and a similiar expression can be found for the other boundary.

Remarks

- The choice of boundary conditions is dictated by the physics and mathematical well-posedness. However, the discrete version depends on numerical stability considerations. ALWAYS CHECK FOR STABILITY AT ENDPOINTS.

- This is not a theorem, but in general you want to use the same order of finite difference approximation to the boundary conditions as you did for the interior points. What happens if you go with lower order? You defeat the purpose of using a high order scheme. What happens when you go higher order? In general, you get an ill-poosed linear algebraic problem.
- Using one-sided derivatives is ok, but these are prone to generate ill-posed linear algebraic problems if you use mix them. This will be apparent when you do your stability study at the end points.

Note on Advection-Diffusion Equation (UPWINDING IS IMPORTANT)

We're not considering this important equation, other than by an example that illustrates one of the important facets that makes this equation tricky to approximate. We'll take as example the forward- time/central-space scheme.

There are a couple of things we're going to learn: upwinding, and also mixing explicit and implicit methods when you have you're solving a problem of mixed type.

We'll work by example:

$$\frac{u_m^{n+1}-u_m^n}{k}+a\frac{u_{m+1}^n-u_{m-1}^2}{2h}=b\frac{u_{m+1}^n-2u_m^n+u_{m-1}^n}{h^2} \tag{152}$$

as an approximation to

$$u_t+au_x=bu_{xx},\ \text{take}\ \overbrace{a>0}^{\text{for concreteness}}\ \text{and}\ \overbrace{b>0}^{\text{must be}} \tag{153}$$

not because its a good algorithm, but because it illustrates the point to be made very simply:

Scheme (152) is $(1,2)$ order accurate and 2^{nd} order accurate overall because of stability requirement

$$b\mu\le\frac{1}{2}\ \text{(check this!)}$$
$$\text{where}\ \mu=\frac{h}{k^2}$$

Note that this restriction is rather severe on the time step, since getting good resolution in space by making h small might mean really tiny k, which means long computing times.

Now,

$$\text{let } \alpha \equiv \frac{ha}{2b} = \frac{a\lambda}{2b\mu} \quad \text{“Cell Reynold's Number” or“Cell Peclet Number,” and let } \lambda = \frac{k}{h}.$$

The α is a ratio of the importance of inertial (wave-like or signal-propagating effects to diffusion effects). So that if $\alpha < 1$ we have a diffusion dominated problem, etc.

Solving for u_m^{n+1} and taking the absolute value of both sides yields

$$|u_m^{n+1}| \leq (1 - 2b\mu)|u_m^n| + b\mu(1 - \alpha)|u_{m+1}^n| + b\mu(1 + \alpha)|u_{m-1}^n|$$

Since we require that

$$\max_m |u_m^{n+1}| \leq \max_m |u_m^n| \tag{154}$$

for parabolic equation approximations, by setting $\alpha \leq 1$ we satisfy (154).

Hence we need to satisfy two conditions

$$\begin{cases} b\mu \leq \frac{1}{2} \\ \alpha \leq 1 \leftarrow \text{ satisfied if } h \leq \dfrac{2b}{a} \end{cases}$$

the second is a restriction on mesh spacing and could be very restrictive.

If $b\mu \leq \dfrac{1}{2}$ is the stability condition, what does $\alpha \leq 1$ do? It guarantees that the scheme will behave qualitatively like a parabolic equation approximation. What you will see if you set $\alpha > 1$ is the appearance of spurious oscillations ... they usually do not grow excessively and they result from inadequate resolution.

One way to avoid the restriction on mesh-spacing is to use “upwind differencing.” The scheme then is

$$\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_m^n - u_{m-1}^n}{h} = b\frac{1}{h^2}\left(u_{m+1}^b - 2u_m^n + u_{m-1}^n\right) \tag{155}$$

or $u_m^{n+1} = [1 - 2b\mu(1+\alpha)]u_m^n + b\mu u_{m+1}^n + b\mu(1+2\alpha)u_{m-1}^n$

If $1 - 2b\mu(1+\alpha) \to 0 \Rightarrow$ scheme satisfies (154). This is satisfied if

$$2b\mu + a\lambda \leq 1$$

Oscillations are eliminated but now we have 1-order accuracy in space. When b small and a large (typical) this condition is much less restrictive than $h \leq 2b/a$. Note, however that (155) can be written as

$$\frac{u_m^{n+1} - u_m^n}{k} + a\frac{u_{m+1-u_{m-1}}^n}{2h} = \underbrace{\left(b + \frac{ah}{2}\right)}_{\text{"artificial viscosity" added to make solutions non- oscillatory.}} \left(\frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{h^2}\right)h^2$$

Remark

- Note that upwinding direction was given by a. If $a < 0$, the derivative would involve $m+1$ and m values of u instead.
- What if a has different signs for different parts of the domain? Use a switching in your code to check for upwinding direction and then change the derivative to the relative upwind direction.
- One way to avoid oscillations due to constraints on the Peclet number and at the same time get larger time steps for asymptotic stability is to compute the problem by using an explicit upwind in the advective term U_x, which will generate a mild restriction on mesh spacing h, and then go implicit in the diffusive term $U_x x$, such as Crank Nicholson. The resulting code will be quite robust.

0.5.3 Reduction of Parabolic Equations to a System of ODE's

$$\text{Consider} \begin{cases} \dfrac{\partial U}{\partial t} = \dfrac{\partial^2 U}{\partial x^2} \quad 0 < x < X \qquad t > 0 \\ U(x,0) = g(x) \qquad\qquad\qquad 0 \le x \le X \\ \text{and known BV's at } x = 0 \quad \text{and } x = X \quad \forall t > 0 \end{cases}$$

Semi-discretizing, using center differences in x (as a particular example)

$$\frac{dU(t)}{dt} = \frac{1}{h^2}\Big\{U(x-h,t) - 2U(x,t) + U(x+h,t)\Big\} + O(h^2)$$

Subdivide interval $0 \le x \le X$ into N equal subintervals with $x_i = ih$, $i = 0 \cdots N$, where $Nh = X$. $u_i(t)$ is an approximation to $u_i(t)$, where $U_i = u(ih,t)$ so the $i = 0$ and $i = N$ are boundary lines. The equation for u_i at some time t are

$$N-1 \text{ ODE's} \begin{cases} \frac{du_1(t)}{dt} = \frac{1}{h^2}(u_0 - 2u_1 + u_2) \\ \frac{du_2(t)}{dt} = \frac{1}{h^2}(u_1 - 2u_2 + u_3) \\ \vdots \qquad \vdots \\ \frac{du_{N-1}}{dt} = \frac{1}{h^2}(u_{N-2} - 2u_{N-1} + u_N) \end{cases}$$

u_0 and u_N are known values (boundary values) let $\mathbf{V}(t) = [u_1, u_2 \dots u_{N-1}]^T$ then the system can be written as

$$\frac{d\mathbf{V}(t)}{dt} = A\mathbf{V}(t) + \mathbf{b} \tag{156}$$

$\mathbf{b}$ is a column of zero's and known values of u

$$A = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 1 & -2 \end{bmatrix} \quad (N-1) \times (N-1) \text{ matrix}$$

Remark: for $y = y(t)$, the solution to

$$\frac{dy}{dt} = ay + b$$
$$y(0) = g$$
$$\text{is} \quad y(t) = -\frac{b}{a} + \left(g + \frac{b}{a}\right) e^{at}$$

□

Solution to

$$\mathbf{V}(t) = -A^{-1}\mathbf{b} + e^{tA}(\mathbf{g} + \mathbf{A^{-1}b}) \tag{157}$$

$$\therefore \quad \mathbf{V}(t+k) = -A^{-1}\mathbf{b} + e^{kA}e^{tA}(\mathbf{g} + \mathbf{A^{-1}b}) \tag{158}$$

substitutions (157) in (158)

$$\mathbf{V}(t+k) = -A^{-1}\mathbf{b} + e^{kA}\left(\mathbf{V}(t) + A^{-1}\mathbf{b}\right)$$

Note: if $\mathbf{b} = 0 \Rightarrow \mathbf{V}(t+k) = e^{kA}\mathbf{V}(t)$.

Stability: perturb g to g^* then

$$\mathbf{V}^*(t) = A^{-1}\mathbf{b} + e^{tA}(\mathbf{g}^* + A^{-1}\mathbf{b})$$

subtracting $\underbrace{\mathbf{V}^*(t) - \mathbf{V}(t)}_{\mathbf{e(t)}} = e^{TA}\underbrace{(\mathbf{g}^* - \mathbf{g})}_{\mathbf{e(0)}}$

$$\mathbf{e}(t) = e^{tA}\mathbf{e}(0)$$

so we require that $||A|| \leq 1$ for stability.

Note on $\dfrac{dV}{dt} = A\mathbf{V} + \mathbf{b}$

Take P a constant coefficient real $n \times n$ matrix, then

$$e^P = I + P + \frac{P^2}{2} + \frac{P^3}{3!} + \cdots \sum_{m=0}^{\infty} \frac{P^m}{m!} \tag{159}$$

here $P^0 \equiv I$ is the $n \times n$ identity matrix

If Q is a real $n \times n$ matrix such that $PQ = QP$ (commute) then

$$e^P e^Q = e^Q e^P = e^{P+Q}$$

Hence $e^P e^{-P} = e^{-P} e^P = e^0 = I$

premultiplication of $e^P e^{-P} = I$ by $(e^P)^{-1}$ then shows that

$$e^{-P} = (e^P)^{-1}$$

On putting $P = At$ in (159) and differentiating with regards to t we get that

$$\frac{d}{dt}(e^{At}) = Ae^{At} = e^{At}A$$

Now consider $\mathbf{V}(t) = e^{At}\mathbf{g}$ where $\mathbf{g}$ is independent of t. This clearly satisfies the condition $\mathbf{V}(0) = \mathbf{g}$. Differentiation with regards to t gives

$$\frac{d\mathbf{V}}{dt} = Ae^{At}g = A\mathbf{V}$$

In other words the solution of

$$\frac{d\mathbf{V}}{dt} = A\mathbf{V} \quad \text{with } \mathbf{V}(0) = \mathbf{g} \quad , \quad \text{is}$$
$$\mathbf{V}(t) = e^{At}\mathbf{g}$$

Similarly, the vector function $\begin{cases} \mathbf{V}(t) = -A^{-1}\mathbf{b} + e^{tA}(\mathbf{g} + A^{-1}\mathbf{b}) \\ \mathbf{V}(0) = \mathbf{g} \end{cases}$

is the solution of $\dfrac{d\mathbf{V}}{dt} = A\mathbf{V} + \mathbf{b}$

provided b and A are independent of t

Finite Difference Schemes from Systems of ODE's

For simplicity assume g given and that the boundary values associated with $\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}$ are 0.

$$\text{for } 0 \leq x \leq X$$

The solution is approximately

$$\mathbf{V}(t+k) = e^{kA}\mathbf{V}(t) \quad t = 0, k, 2k, \ldots \tag{160}$$

where A is given before. The FD comes in approximating e^{kA}. First, notice that

$$e^{kA} = I + kA + \frac{1}{2}k^2A^2 + \cdots$$

Then, an obvious approximation is $e^{kA} \approx I + kA$, so (160) is approximately

$$\mathbf{V}(t+k) = (I + kA)V(t) \tag{161}$$

if $t = nk$ and $\mu = k/h^2$ then (161) is

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ : \\ : \\ u_{M-1}^{n+1} \end{bmatrix} = \begin{bmatrix} (1-2\mu) & \mu & 0 & & \\ \mu & (1-2\mu) & \mu & & \\ & \ddots & \ddots & \ddots & \\ & \mu & (1-2\mu) & \mu & \\ & 0 & \mu & (1-2\mu) & \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ : \\ : \\ u_{M-1}^n \end{bmatrix}$$

or $u_m^{n+1} = \mu u_{m-1}^n + (1-2\mu)u_m^n + \mu u_{m+1}^2, \quad m = 1, 2 \ldots M-1$

$$\text{with } Mh = X$$

We can use Padé Approximants to get better approximations to e^{kA}.

Padé Approximants to e^θ , where θ is real:

Assume e^θ can be approximated as $\dfrac{1+p_1\theta}{1+q_1\theta}$ $\quad ; p_1, q_1$ constants.

then we need 2 equations to determine p_1, q_1.

$$e^\theta = \frac{1+p_1\theta}{1+q_1\theta} + c_3\theta^3 + c_4\theta^4 + \cdots$$

multiplying both sides by denominator

$$\therefore (1+q_1\theta)(1+\theta+\frac{1}{2}\theta^2+\frac{1}{6}\theta^3\cdots) = 1 + p_1\theta + (1+q_1\theta)(c_3\theta^3 + c_4\theta^4 + \cdots)$$

Hence $(1+q_1-p_1)\theta + (\frac{1}{2}+q_1)\theta^2 + \left(\frac{1}{6}+\frac{1}{2}q_1 - c_3\right)\theta^3 +$ higher order terms $= 0$

This is uniquely satisfied to $\mathcal{O}(\theta^3)$ by $p_1 = \dfrac{1}{2} \quad q_1 = -\dfrac{1}{2} \quad c_3 = -\dfrac{1}{12}$

Hence $\hat{r}_{1/1} \equiv \dfrac{1+\frac{1}{2}\theta}{1-\frac{1}{2}\theta}$ is a $(1,1)$ Padé Approximation of e^θ of order 2. It has leading-order error $= -\dfrac{1}{12}\theta^3$.

In general

$$e^\theta = \frac{1+p_1\theta+p_2\theta^2+\cdots p_T\theta^T}{1+q_1\theta+q_2\theta^2+\cdots q_s\theta^s} + \underbrace{c_{S+T+1}}_{\text{constant.}}\ \theta^{S+T+1} + \mathcal{O}\left(\theta^{S+t+2}\right)$$

Hence $\hat{r}_{1/S} = \dfrac{P_T(\theta)}{Q_S(\theta)}$ is the (T,S) Padé Appointment of order $T+S$ to e^θ

Exercise: Show that

(T,S)	$\hat{r}_{T/S}$	Principal Error Term
$(1,0)$	$1+\theta$	$\frac{1}{2}\theta^2$
$(2,0)$	$1+\theta+\frac{1}{2}\theta^2$	$\frac{1}{6}\theta^3$
$(2,1)$	$\frac{1+\frac{2}{3}\theta+\frac{1}{6}\theta^2}{1-\frac{1}{3}\theta}$	$-\frac{1}{72}\theta^4$

□

Example:

Approximate $\mathbf{V}(t+k) = e^{kA}\mathbf{V}(t)$ using $\hat{v}_{1/1}$

$$\Rightarrow \mathbf{V}(t+k) = (I-\frac{1}{2}kA)^{-1}(I+\frac{1}{2}kA)\mathbf{V}(t)$$

or $(I - \frac{1}{2}kA)\mathbf{V}(t+k) = (I + \frac{1}{2}kA)\mathbf{V}(t)$

or $-\mu u_{m-1}^{n+1} + 2(1+\mu)u_m^{n+1} - \mu u_{m+1}^{n+1} = \mu u_{m-1}^n + 2(1-\mu)u_m^n + \mu u_{m+1}^n m = 1 \cdots M-1$

Crank-Nicholson!

The $\hat{r}_{1/1}$ is also called a "Unitary" approximation, which is important property of the Schroedinger equation which is important to preserve.

A and L Stability

Continuing our discussion of $\dfrac{\partial U}{\partial t} = \dfrac{\partial^2 U}{\partial^2 x} \quad t > 0 < X$

with $u(0,x) = g(x)$ and assume for simplicity that boundary values are zero.

Take $\mathbf{V}(0) = [g_1, g_2 \cdots g_{M-1}]^T$ assume that $Mh = X$

$\mathbf{V}(t)$ is an approximating vector to $\mathbf{U}(t)$

$$\mathbf{V}(t_n + k) = \hat{r}_{T/S}(k\mathbf{A})\mathbf{V}(t_n)$$

but

$$\mathbf{V}(t_n) = \hat{r}_{T/S}(k\mathbf{A})\mathbf{A}(t_{n-1}), \text{and so on,}$$

which leads recursively to

$$\mathbf{V}(t_n) = [\hat{r}_{T/S}(kA)]^n \mathbf{V}(0) \tag{162}$$

The eigenvalues of $A\Big($for center difference approx to $\dfrac{\partial^2}{\partial x^2}\Big)$ are

$$-\frac{4}{h^2}\sin^2\left(\frac{\ell\pi}{2M}\right) \quad s = 1, 2, \cdots M-1$$

and are all different. Hence the eigenvectors of A are independent and a basis for the $(M-1)$-dimensional space of the vector g of initial values.$\therefore$

$$\mathbf{g} = \sum_{\ell=1}^{M-1} c_\ell \phi_\ell$$

$\therefore$ (162) can be expressed as

$$\mathbf{V}(t_n) = [\hat{r}_{T/S}(kA)]^n \sum_{l=1}^{M-1} c_l \phi_l = \sum_{l=1}^{M-1} c_l [\hat{r}_{T/S}(kA)]^n \phi_l \tag{163}$$

Since $A\phi_l = \lambda_l\phi_l$ and we know that $f(A)\phi_l = f(\lambda_l)\phi_l$ it follows that (163) can be expressed as

$$\mathbf{V}(t_n) = \sum_{l=1}^{M-1} c_l[\hat{r}_{T/S}(k\lambda_l)]^n\phi_l. \tag{164}$$

(164) shows that $\mathbf{V}(t_n)$ will tend to the null vector as $n \to \infty$ if and only if

$$|\hat{r}_{T/S}(k\lambda_l)| < 1 \quad l = 1, 2, \cdots M-1$$

If this condition is subject to $\mu = k/h^2$ value, the equations are "CONDITIONALLY STABLE."

When $|\hat{r}_{T/S}(\lambda_l k)| < 1 \quad \forall \quad \mu > 0 \Rightarrow$ "$A-$ Stable" or unconditionally stable

Although A stability implies that $-1 < \hat{r}_{T/S}(k\gamma_s) < 1$ for real $\hat{r}_{T/S}$, it is possible that some values of $\hat{r}_{T/S(k\lambda_l)}$ be close to -1 and hence for these $\hat{r}_{T/S}(k\lambda_l)$ will alternate in sign as n increases and diminish in amplitude only very slowly. This phenomenon is particularly pronounced in the x-neighborhoods of points of the discontinuity either in the initial values or between boundary and intial values.

The real coefficients of $\hat{r}_{T/S}$ would clearly be free of unwanted oscillations if $0 < \hat{r}_{T/S} < 1$ and $\hat{r}_{T/S}(k\lambda_l) \to 0$ monotonically as $k\lambda_l$ increase in magnitude. The $(0,1)$ Padé Approximant

$\hat{r}_{0/1} = \dfrac{1}{1-k\lambda_l}$, λ_l real negative would clearly have this property. This corresponds to implicit (backwards) Euler.

If $|\hat{r}_{T/S}(k\lambda_\ell)| < 1$ for $\quad l = 1, \cdots M-1$ we say the scheme is L_0 stable

For Crank-Nicholson $\hat{r}_{1/1}(-z) = \dfrac{1-\frac{1}{2}z}{1+\frac{1}{2}z} = \dfrac{2/z-1}{2/z+1}$

$|\hat{r}_{1/1}(-z)| < 1 \quad \forall z > 0$ but $\hat{r}_{1/1}(-z) \to -1$ as $z \to \infty$

$\therefore$ CN is $A-$ stable.

In order to avoid unwanted oscillations one can show that it is sufficient for $c_{m-1}\phi_{m-1}$ to decay to zero faster than the lowest component $c_1\phi_1$. This implies that

$$\boxed{\frac{k}{h} < \frac{x}{\pi}}$$

see Lawson, Morris (1978) J Num Anal SIAM 15, pp 1212-25.

0.6 HIGHER-ORDER EVOLUTION EQUATIONS AND SPLIT-STEP METHODS

We mean higher-order equations in time. The most common are 2^{nd} order equations. Example:

$$U_{tt} - a^2 U_{xx} = 0$$

the wave-equation, which admits a solution composed of a right-going and left-going wave. It belongs to the more general 2-order hyperbolic family of equations

$$U_{tt} + 2bU_{tx} = a^2 U_{xx} + cU_x + dU_t + eU + f(t,x) \text{where } b^2 < a^2$$

$$\text{Take } \begin{cases} U_{tt} - a^2 U_{xx} = 0 \\ U(0,x) = U_0(x) \\ U_t(0,x) = U_1(x) \end{cases}$$

has a general solution

$$U(x,t) = \frac{1}{2}\Big[U_0(x-at) + U_0(x+a_t)\Big] + \frac{1}{2a}\int_{x-at}^{x+at} U_1(y)dy$$

As you can see, the solution is composed of a left-going and right-going wave. The above problem can also be cast as a system of equations:

Example

On $-1 \le x \le 1 \quad t \ge 0$

Take $\quad U_{tt} = U_{xx}$. It is equivalent to solving

$$\begin{cases} U_t + V_x = 0 \\ V_t + U_x = 0 \quad \text{on } 0 \le x \le 1, \quad t \ge 0 \end{cases}$$

So we can let $\mathbf{V} = (u, v)^T$ and solve

$$\frac{\partial \mathbf{V}}{\partial t} + A\frac{\partial \mathbf{V}}{\partial x} = \mathbf{0} \quad \text{where } A \text{ is diagonalizable and has only real e'values}$$

□

Example Another higher-order equation which appears with some regularity: The Euler-Bernouilli Equation $U_{tt} = -b^2 U_{xxxx}$ also has a general solution composed of 2 basic solutions. This equation is neither parabolic or hyperbolic ... the solution does not become smoother as $t \to \infty$, like parabolic equations, nor does the solution have finite speed of propagation as it does for hyperbolic equations.

□

We will not consider in detail the general solution of higher-in-time PDE's. Merely, we indicate the general technique for their solution. A sensible technique is to turn the 2nd-order (or higher order) equation into a system of 1^{st}-order equations, and then, after projecting or discretizing in space, use ODE theory to find the best time integration algorithm for the resulting problem. Caution: Make sure that you engineer the eventual linear-algebraic problem in a compact way. If you use the above trick you will get very large and sparse matrices with large bandwidths. You can most likely turn the sparse large band width matrix problem into a sparse small bandwidth problem using careful computer engineering practices. We'll see how this is done in a number of examples considered in this section and in the Elliptic Equation Section.

Splitting and ADI. Nonlinear Problems and Problems in Several Space Dimensions

Splitting techniques can be used to efficiently solve certain nonlinear evolution problems and equations in several space dimensions. Consider the following example:

$$\frac{\partial U}{\partial t} = \nabla(a\nabla U) + f \quad 0 \le x, y \le 1$$
$$t > 0$$
$$a = a(x, y) \text{ is bounded in } [0, 1] \times [0, 1]$$

For simplicity, assume that the grid spacing in x and y is the same: define such grid spacing by Δx. A naive discretization in space leads to

$$u'_{k,\ell} = \frac{1}{(\Delta x)^2}\Big[a_{k-\frac{1}{2},\ell}u_{k-1,\ell} + a_{k,l-\frac{1}{2}}u_{k,\ell-1} + a_{k+\frac{1}{2},\ell}u_{k+1,\ell}a_{k,\ell+\frac{1}{2}}u_{k\ell+1}$$
$$-\left(a_{k-\frac{1}{2},\ell} + a_{k,\ell-\frac{1}{2}} + a_{k+\frac{1}{2},\ell}u + a_{k,\ell+\frac{1}{2}}\right)u_{k,\ell}\Big] + p_{k,\ell} + f_{k,\ell}$$
with $k, \ell = 1 \dots d$
and $u' \equiv \dfrac{\partial}{\partial t}u$

with $k, \ell = 1 \dots d$

p and f are the boundary and forcing term contributions, assumed to be dependent of time. Assume they are 0 for now. In this case the above equation is of the form

$$\mathbf{u}' = \frac{1}{\Delta x}(B_x + B_y)\mathbf{u} \quad t \geq 0, \quad \mathbf{u}(0) \text{ given} \tag{165}$$

B_x and B_y are $d^2 \times d^2$ matrices approximating differential operators in x and y.

B_y is a block-diagonal matrix and its diagonal is constructed from the tri-diagonal $d \times d$ matrices:

$$\begin{bmatrix}
-\left(b_{\frac{1}{2}} + b_{\frac{3}{2}}\right) & b_{\frac{3}{2}} & 0 & \cdots & & 0 \\
b_{\frac{3}{2}} & -\left(b_{\frac{3}{2}} + b_{\frac{5}{2}}\right) & b_{\frac{5}{2}} & 0 & \cdots & 0 \\
: & \ddots & \ddots & \ddots & & : \\
0 & & 0 & b_{d-\frac{3}{2}} & -\left(b_{d-\frac{3}{2}} + b_{d-\frac{1}{2}}\right) & b_{d-\frac{1}{2}} \\
0 & & \cdots & 0 & b_{d-\frac{1}{2}} & -\left(b_{d-\frac{1}{2}} + b_{d+\frac{1}{2}}\right)
\end{bmatrix}$$

where $b_\ell = a_{k,\ell} \quad k = 1, 2, \cdots d$

B_x contains all remaining terms.

Its sparsity pattern is block-tridiagonal provided the grid is ordered by rows rather than by columns. Any other ordering will lead to a sparse but large bandwidth system.

The formal solution of (165) is

$$u^{n+1} = e^{\mu(B_x+B_y)}u^n \quad n \geq 0$$

Now we use a Padé approximant to discretize the matrix operators: for example using $\hat{r}_{1/1}$ to approximate the exponential operator, leads to

$$\mathbf{u}^{n+1} = \Big[I - \frac{1}{2}\mu(B_x + B_y)\Big]^{-1}\Big[I + \frac{1}{2}\mu(B_x + B_y)\Big]\mathbf{u}^n \tag{166}$$

($CrankNicholson$ in $2D$)

The following identity is true for matrices:

$$e^{t(Q+S)} = e^{tQ}e^{tS} \quad t \geq 0 \tag{167}$$

ONLY IF Q and S COMMUTE. Q and S are square and of equal dimension. Assume they do, then:

$$\begin{aligned}\mathbf{u}^{n+1} &= \hat{r}_{1/1}(\mu B_x)\hat{r}_{1/1}(\mu B_y)\mathbf{u}^n \\ &= \underbrace{\left(I - \frac{1}{2}\mu B_x\right)^{-1}}_{I}(I + \frac{1}{2}\mu B_x)(I - \frac{1}{2}\mu B_y)\underbrace{\left(I + \frac{1}{2}\mu B_y\right)}_{II}\mathbf{u^n}\end{aligned} \tag{168}$$

here $n \geq 0$

The advantage of solving (168) over (166) is that inspite of having to solve 2 linear systems at each time step, with $I - \frac{1}{2}\mu B_y$ tridiagonal, and $I - \frac{1}{2}\mu B_x$ tridiagonal is that it can solve (168) in $O(d^2)$ operations!

Identity (167) would be true in this case if $a = 1$. In general, the identity is only approximately true:

$$e^{tQ}e^{tS} - e^{t(Q+S)} = (I + tQ + \frac{1}{2}t^2Q^2 + \cdots)(I + tS + \frac{1}{2}t^2S^2 + \cdots)$$
$$- \left\{I + t(Q+S) + \frac{t^2}{2}(Q+S)^2 + \cdots\right\} = \frac{1}{2}t^2[Q,S] + O(t^3)$$

where $[Q,S] \equiv QS - SQ$ the cummutator

$\therefore$ if $[B_x, B_y] \neq 0$

then $e^{\mu B_x}e^{\mu B_y} - e^{u(Bx+By)} = 0(\mu^2)$

if μ is sufficiently small, the approximation is fruitful.

Exercise

Show that "Strang Splitting"

$$e^{\mu(B_x+B_y)} \approx e^{\frac{1}{2}\mu B_x} e^{\mu B_y} e^{\frac{1}{2}\mu Bx} + O(\mu^3)$$

□

What to do if boundary conditions and/or forcing non-zero?

Take

$$\mathbf{u}' = \frac{1}{(\Delta x)^2}(B_x + B_y)\mathbf{u} + \mathbf{h}(t),\ t \geq 0$$
$$\mathbf{u}(0) \text{ given}$$

then

$$\mathbf{u}^{n+1} = e^{\mu(B_x+B_y)}\mathbf{u}^n + \Delta t \int_0^1 e^{(1-\tau)\mu(B_x+B_y)} \mathbf{h}\left((n+\tau)\Delta t\right) d\tau$$

$n \geq 0$, and replace integral by trapezoidal rule:

$$\mathbf{u}^{n+1} = e^{\mu(Bx+By)}\left[\mathbf{u}^n + \tfrac{1}{2}\Delta t\, \mathbf{h}(n\Delta t)\right] + \tfrac{1}{2}\Delta t\, \mathbf{h}\left((n+1)\Delta t\right)$$

Example Can use the split-step to efficiently solve the Nonlinear Schrodinger Equation (NLS). First, we use $\hat{r}_{1/1}$ since it is a unitary approximation and commutes with the Schrodinger operator. This is important in quantum mechanics.

Take

$$u_t = i\nabla^2 u + i\sigma|u|^2 u \qquad u(x,y,t) \in \mathbb{C}$$

Let $\mathcal{L}_1 = i\nabla^2 \quad \mathcal{L}_2 u = i\sigma|u|^2 u$. Here, $\nabla^2 = \partial_{xx} + \partial_{yy}$, in two space dimensions.

1. Advance linear part of NLS

$$u_t = i\nabla^2 u,$$

i.e. using $\mathcal{L}_1$ and Fourier Transform, so that the k^{th} spectral component advances as

$$\hat{u}_k\left(t+\frac{\Delta t}{2}\right) = e^{-ik^2\frac{\Delta t}{2}}\hat{u}_k(t)$$

using an FFT. Then take inverse FFT to obtain a quantity called $\bar{\mathbf{u}}\,(t+\Delta t/2)$.

2. To propagate under $\mathcal{L}_2$, solve

$$\bar{\bar{\mathbf{u}}}_{\mathbf{t}} = \mathbf{i}\sigma|\bar{\mathbf{u}}|^{\mathbf{2}}\bar{\mathbf{u}}$$

which has an exact solution since $|u|^2$ is conserved. The solution is

$$\bar{\bar{\mathbf{u}}}(\mathbf{t}+\mathbf{\Delta t}) = \mathbf{e}^{\mathbf{i}\sigma\left|\bar{\mathbf{u}}\left(\mathbf{t}+\frac{\mathbf{\Delta t}}{\mathbf{2}}\right)\right|^{\mathbf{2}}\mathbf{\Delta t}\bar{\mathbf{u}}\left(\mathbf{t}+\frac{\mathbf{\Delta t}}{\mathbf{2}}\right)}.$$

3. The final stage is another half-step propagation under $\mathcal{L}_1$.

$$\hat{u}_k(t+\Delta t) = e^{-ik^2\frac{\Delta t}{2}}\hat{\bar{\mathbf{u}}}_{\mathbf{k}}(\mathbf{t}+\mathbf{\Delta t}).$$

then inverse FFT of $\hat{\mathbf{u}}_k$ gives final "solution" after a single time step. Method requires 4 FFT's and 1 experimentation/time step and is $0(\Delta t^3)$ accurate. It is expensive if you do not use FFT's. See Tappert for more details.

□

ADI (Alternating Implicit Direction) Methods

A splitting method. Take

$$U_t = b_1U_{xx} + b_2U_{yy}$$

on a unit square.

Let $A_1U = b_1U_{xx} \quad A_2U = b_2U_{yy}$

$$\text{then } U_t = A_1U + A_2U$$

as before, supposing we used $\hat{r}_{1/1}$, with $k \equiv$ time step:

$$\left(I - \frac{k}{2}A_1, -\frac{k}{2}A_2\right)\mathbf{u}^{n+1} = \left(I + \frac{k}{2}A_1 + \frac{k}{2}A_2\right)\mathbf{u}^n + \mathcal{O}(b^3)$$

Since $(1 \pm a_1)(1 \pm a_2) = 1 \pm a_1 \pm a_2 + a_1 a_2$

we add $\dfrac{k^2 A_1 A_2}{4} \mathbf{u}^{n+1}$ to both sides

$$\left(I - \frac{k}{2}A_1 - \frac{k}{2}A_2 + \frac{k^2}{4}A_1A_2\right)\mathbf{u}^{n+1} =$$
$$\left(I + \frac{k}{2}A_1 + \frac{k}{2}A_2 + \frac{k^2}{4}A_1A_2\right)\mathbf{u}^n + \frac{k^2}{4}A_1A_2(\mathbf{u}^{n+1} - \mathbf{u}^n) + \mathcal{O}(k^3)$$

which can be factored

$$\left(I - \frac{k}{2}A_1\right)\left(I - \frac{k}{2}A_2\right)\mathbf{u}^{n+1} = \left(I + \frac{k}{2}A_1\right) + \left(I + \frac{k}{2}A_2\right)\mathbf{u}^n + \frac{k^2}{4}A_1A_2\underbrace{(\mathbf{u}^{n+1} - \mathbf{u}^n)}_{\mathcal{O}(k)}\ \underbrace{\mathcal{O}(k^3)}_{0(k^3)}$$

If A_1 and A_2 are discretized using the 2^{nd} order stencil, we obtain tridiagonal matrices that are sparse and easily solved. Let A_{1h} and A_{2h} be 2^{nd} order approximations. Then

$$\begin{aligned}\left(I - \frac{k}{2}A_{1h}\right)\left(I - \frac{k}{2}A_{2h}\right)\mathbf{u}^{n+1} &= \left(I + \frac{k}{2}A_{1h}\right)(I + \frac{k}{2}\Delta_{2h})\mathbf{u}^n \\ &+ \mathcal{O}(k^3) + \mathcal{O}(kh^2)\end{aligned}$$

or

$$\left(I - \frac{k}{2}A_{1h}\right)\left(I - \frac{k}{2}A_{2h}\right)\mathbf{u}^{n+\frac{1}{2}} = \left(I + \frac{k}{2}A_{1h}\right)\left(I + \frac{k}{2}A_{2h}\right)\mathbf{u}^n \tag{169}$$

The Peaceman-Ratchford Algorithm to solve (169)

$$\begin{cases} \left(I - \frac{k}{2}A_{1h}\right)\tilde{\mathbf{u}}^{n+\frac{1}{2}} = \left(I + \frac{k}{2}A_{2h}\right)u^n \\ \left(I - \frac{k}{2}A_{2h}\right)u^{n+1} = \left(I + \frac{h}{2}A_{1h}\right)\tilde{\mathbf{u}}^{n+\frac{1}{2}} \\ \text{(unconditionally stable).} \end{cases}$$

And we can see why it is called ADI $\cdots$

Another algorithm to solve (169):

Douglas-Rachford Algorithm: $(1,2)$ scheme, unconditionally stable.

$$\begin{cases} \left(I - \frac{k}{2}A_{1h}\right)\tilde{\mathbf{u}}^{n+\frac{1}{2}} = \left(I + \frac{k}{2}A_{2h}\right)\mathbf{u}^n \\ \left(I - \frac{k}{2}A_{2h}\right)u^{n+\frac{1}{2}} = \tilde{\mathbf{u}}^{n+\frac{1}{2}} - kA_{2h}\mathbf{u}^n \\ \quad \text{(unconditionally stable).} \end{cases}$$

Implementation Comments:

1. ADI schemes require intermediate values on the boundary. The approximation must be chosen so that no instabilities are introduced. Start with something simple.
2. The code can be made very fast if the row-column switching discussed previously is implemented.
3. Higher-order matrix representations or non-sparse representations may require an iterative technique for solution: for most cases an SOR or Conjugate Gradient (if symmetric) are quite efficient.

□

0.7 ELLIPTIC EQUATIONS

Brief overview Refer to Figure 30 Archetypical Equation in 2D:

$$\nabla^2 U = f(x,y) \quad \text{in a domain } \Omega(x,y).$$

with $\nabla^2 \equiv \partial_{xx} + \partial_{yy}$

is called Poisson's Equation. If $f = 0$ we call it Laplace's equation. The solutions to Laplace's equations are called harmonic functions and are intimately tied to the theory of complex analysis.

Boundary Conditions:

(1) (field specified at boundary) Dirchlet: $U = b_1$ on $\partial\Omega$

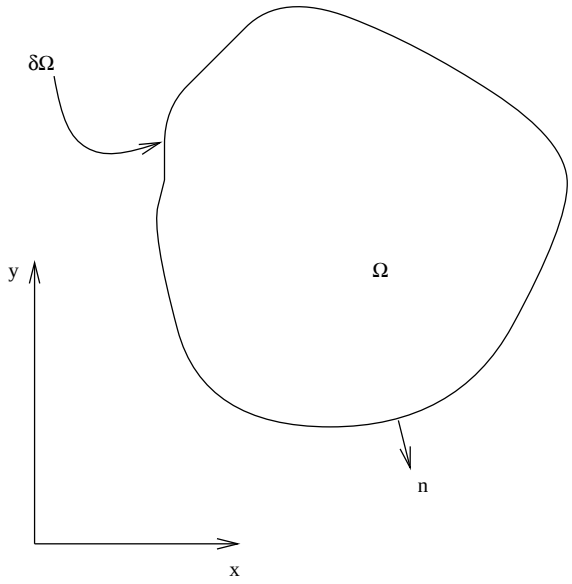

Figure 30: Domain of definition for Poisson's Problem in 2 dimensions. The $\hat{n}$ indicates the convention on the unit normal to the boundary $\partial\Omega$.

(2) Neumann: $\dfrac{\partial U}{\partial n} \equiv \hat{n} \cdot \nabla U = b_2$ on $\partial\Omega$

(3) Robin: mix of the two above: $\dfrac{\partial U}{\partial n} + \alpha U = b_3$ on $\partial\Omega$. Only 1 boundary condition may be specified on $\partial\Omega$. But can have Neumann, say, for one portion of $\partial\Omega$ and Dirichlet for the other, for example.

Only 1 boundary condition may be specified on $\partial\Omega$. But can have Neumann, say, for one portion of $\partial\Omega$ and Dirichlet for the other, for example.

Physically: Poisson equation describes many things. For example, the steady state-temperatue distribution of an object occupying Ω, with heat sources and sinks represented by f. The Dirichlet boundary conditions represent the situation when the temperature is specified at boundary and Naumann would be if the flux of temperature is specified at boundary. In particular, if $\dfrac{\partial U}{\partial n} = 0|_{\partial\Omega}$ we say we have a perfect insulator boundary.

In order for solution to exist, if Newmann B.C. are specified, is the data must

satisfy the "integrability" condition:

$$\int\int_{\Omega} f dV = \int_{\partial\Omega} b_2 ds$$

(to prove: use divergence theorem).

General Quasi-linear 2nd-order elliptic in 2D has the form:

$$a(x,y)U_{xx} + 2b(x,y)U_{xy} + c(x,y)U_{yy} + d(x,y,U,U_x,U_y) = f(x,y)$$

with $a, c > 0 \quad b^2 < ac$

A 1^{st}-order elliptic equation system example:

$$\begin{cases} U_x - V_y = 0 \\ U_y + V_x = 0 \end{cases} \qquad \text{"Cauchy-Riemann" equations}$$

example of 4^{th} order

$$\nabla^4 U = f \qquad \text{"Biharmonic Equation."}$$

An essential feature of elliptic equation solutions is that they are smoother than the data. For example U has 2 more derivatives then f in the Poisson equation. 4 more than f in biharmonic equation. Solutions to Laplace and Cauchy-Riemann Eqs are infinitely differentiable.

For the general 2^{nd}-order linear constant coefficient elliptic equations, the following "regularity estimate" can be proved:

$$||U||^2_{s+2} \leq C_s \left(||f||^2_s + ||U||^2_0 \right)$$

where

$$||\cdot||^2_s \equiv \sum_{s_1+s_2 \leq s} ||\partial_x^{s_1} \partial_y^{s_2} \cdot ||^2$$

i.e. if solution exists and is finite in L_2, i.e. $||U||_0$ finite and that f has all derivatives of order up to s in $L_2(\mathbb{R}^2) \Rightarrow U$ has $s+2$ in $L_2(R_2)$.

The solution of the elliptic equation is more differentiable than the data and the increase in differentiability = order of equation.

There is an "interior regularity estimate" as well. Suppose $\Omega_1 \in \Omega$ whose boundary is wholly contained in Ω. Then

$$||U||^2_{s+2,\Omega_1} \leq C_s(\Omega, \Omega_1)(||f||^2_{s,\Omega} + ||U||^2_{0,\Omega})$$

For the non-constant coefficient case, we require that coefficients be defined and bounded, and very similar estimates are obtained.

Maximum Principles

These are very important and useful tools in analysis, here, restricted to 2^{nd}-order elliptic equations, although they exist for higher order elliptic equations as well. The 2^{nd} derivative gives information on a functions' extrema $\therefore$ maximum principles are useful tools in the analysis of solutions of 2^{nd} -order elliptic equations. Two theorems:

I. Theorem (Max Value) let $L\phi = a\phi_{xx} + 2b\phi_{xy} + c\phi_{yy}$, $a, c > 0$ and $b^2 < ac$ i.e. L is an elliptic operator. If U satisfies $LU \geq 0$ on a bounded domain $\Omega \Rightarrow U$ has its maximum on $\partial\Omega$.

Remark: In 1D, recall that if $U'' > 0$ on some closed interval in $x \Rightarrow$ max value U is at interval ends (convince yourself).

II. Theorem (Max/Min): If elliptic equation

$$aU_{xx} + 2bU_{xy} + cU_{yy} + d_1U_x + d_2U_y + eU = 0$$

holds on Ω $a, c > 0$, $e \leq 0 \Rightarrow U(x, y)$ cannot have a positive local maximum or a negative local minimum in interior of Ω.

Proof (from Strikwerda): Prove only I when $LU > 0$ and II when $e < 0$. Cases when $LU \geq 0$ and $e \leq 0$ take a little more effort. Assume $U \in C^3$

I: $U \in C^3$ has local max at $(x_0, y_0) \Rightarrow$ gradient of U at(x_0, y_0)

$$U_x(x_0, y_0) = U_y(x_0, y_0) = 0$$

using Taylor's with $U^0_{xx} \equiv U_{xx}(x_0, y_0)$, etc. $\cdots$

$$\begin{aligned} U(x_0 + \Delta x, y_0 + \Delta y) &= U(x_0, y_0) + \frac{1}{2}\left(\Delta x^2 U^0_{xx} + 2\Delta x \Delta y U^0_{xy} + \Delta y^2 U^0_{yy}\right) \\ &+ \mathcal{O}\left(\max(\Delta x, \Delta y)^3\right) \end{aligned}$$

Since $U(x_0 + \Delta x, y_0 + \Delta y) \leq U(x_0, y_0)$ for sufficient small Δx and Δy then

$$\Delta x^2 U^0_{xx} + 2\Delta_x \Delta y U^0_{xy} + 0 y^2 U^0_{yy} \leq 0$$

Since expression is homogeneous of degree 2 in Δx and Δy

$$\alpha^2 U^0_{xx} + 2\alpha\beta U^0_{xy} + \beta^2 U^0_{yy} \leq 0 \tag{170}$$

$$\forall \text{ real } \alpha, \beta.$$

Now, prove I for $LU > 0$. Apply (170) twice. First with $\alpha = \sqrt{a^0} \quad \beta = b^0/\sqrt{a^0}$, and then with $\alpha = 0$ and $\beta^2 = C^0 - (b^0)^2/a^0$, we have

$$\begin{aligned} LU &= a^0 U^0_{xx} + 2b^0 U^0_{xy} + c^0_{yy} \\ &= \left(\sqrt{a^0}\right)^2 U^0_{xx} + 2\sqrt{a^0}\left(\frac{b^0}{\sqrt{a^0}}\right) U^0_{xy} + \left(\frac{b^0}{\sqrt{c^0}}\right)^2 U^0_{yy} + \left(c^0 - \frac{(b^0)^2}{a^0}\right) U^0_{yy} \leq 0 \end{aligned}$$

Since this contradicts assumption that $LU > 0 \Rightarrow$ theorem I is proved.

Proof of Theorem II: only when $e(x, y) < 0$ proof: From Theorem I if U has maximum at (x_0, y_0) then $LU \leq 0 \therefore$

$$-LU(x_0, y_0) = e(x_0, y_0) U(x_0, y_0) \geq 0$$

Since $e < 0 \Rightarrow U(x_0, y_0) \leq 0$ at an interior local maximum. Similarly by considering $-U(x_0, y)$ can show that U is not negative at a local minimum. □

Some uses:

(1) Physical: theorems state that hottest and coldest temps for steady temperature distribution occur at boundaries.

(2) Mathematical: Can use principles to prove uniqueness of solutions to many elliptic equations.

Comments on Boundary Conditions for elliptic equations: Look at Poisson only and consider

$$\begin{aligned} &U = b_1 \text{ on } d\Omega \quad \text{Dirichlet} \\ &\frac{\partial U}{\partial n} = b_2 \text{ on } d\Omega \text{ Neumann} \\ &\frac{\partial U}{\partial U} + \alpha U = b_3 \text{ on } d\Omega \text{ Robin} \end{aligned}$$

if $\partial\Omega$ is smooth, a unique solution exists with dirichlet boundary condition. It also exists for Neumann, if integrability condition is satisfied (note: solution is unique, to within an additive constant).

Some general remarks on local behavior:

(1) If Dirichlet is enforced along smooth portion of boundary $\Rightarrow$ normal derivative at $\partial\Omega$ will be as well behaved as the derivative of the boundary data in the direction of boundary. If boundary data is discontinuous $\Rightarrow$ normal derivative of solution will have unboundedness of discontinuities.

(2) If either Neumann or Robin are enforced at $d\Omega \Rightarrow$ solution differentiable and 1^{st} derivative as well behaved as the boundary data function.

Serious difficulty occurs at points on boundary where boundary condition change from Dirichlet to Newmann or Robin type $\Rightarrow$ one gets unbounded derivatives for u at these points.

(3) Similar difficulties arise in reentrant corners: where local angle is greater than 180°, as measured from inside: second derivative may be unbounded, although solution and 1^{st} derivative bounded (see Figure 31).

In summary: When boundary conditions change type, or when boundary is not smooth, expect derivatives of solution to have unbondedness.

0.7.1 NUMERICAL METHODS FOR THE SOLUTION OF THE POISSON EQUATION

We consider here the solution of the most ubiquitous elliptic equations, the Poisson Equation. One method that will not be given consideration is FEM, the reason being that time constraints will not permit us to do so. We'll consider instead

1) FD methods $\left\{ \begin{array}{c} \text{5-point} \\ \text{9-point} \end{array} \right\}$ for 2D problem

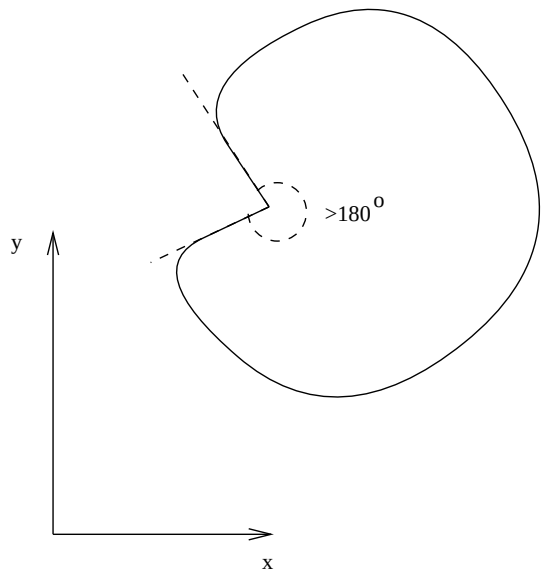

Figure 31: Domain with reentrant corner.

2) Multigrid method → we'll cover only the basics

3) Fast Poisson-Solvers.

We assume that students will have a background in direct and iterative methods for the solution of linear equations and working knowledge of the FFT (See FFT portion of Hw 8..

The 5 and 9-Point Finite Difference Scheme

We limit ourselves to the 2-D Poisson equation

$$\nabla^2 U = f \quad (x,y) \in \Omega$$
$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$
$$f = f(x,y) \text{is a known function}$$

and domain $\Omega \in \mathbb{R}^2$ is bounded, open, connected and has a piecewise smooth boundary $\partial\Omega$.

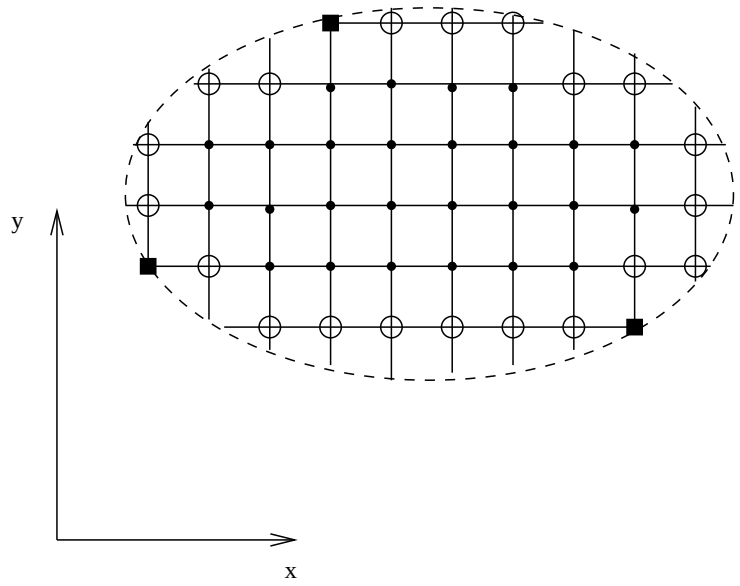

Figure 32: Picture of Ω with $\Omega_{\Delta x}$ indicated. Solid circles indicate "interior" points, hollow circle are "near-boundary" points, filled squares indicate "boundary" points.

Take an arbitrary domain and grid using Δx spacing in BOTH x and y direction. The domain grid is $\equiv \Omega_{\Delta x}$. Gridding will aligned parallel to the x, y coordinate system. The grid is depicted in Figure 32 We need to specify boundary conditions. Consider here Dirichlet boundary condition, for concreteness:

$$U(x,y) = \phi(x,y) \quad \text{for} \qquad (x,y) \in \partial\Omega$$

Take $U_\Delta^{k,l} \equiv U(x_0 + k\Delta x, y_0 + l\Delta x)$, the grid function. We approximate ∇^2 operator acting on the field by center differences to $\mathcal{O}(\Delta x^2)$: the 5-point scheme is thus defined as

$$\frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2\right) u_{k,l} = f_{k,l} \tag{171}$$

where $u_{k,l} = U_\Delta^{k,l} + \mathcal{O}(\Delta x^2)$, an approximation to the grid function.

$$f_{k,l} \equiv f(x_0 + k\Delta x, y_0 + l\Delta x)$$

(171) is written as

$$u_{k-1,l} + u_{k+1,l} + u_{k,1-l} + u_{k,l+1} - 4u_{k,l} = (\Delta x)^2 f_{k,l} \tag{172}$$

computational cell, molecule, or stencil

Of course $\Omega_{\Delta x}$ is comprised of interior, boundary, and near-boundary points. (172) approximates $U_{\Delta x}$ on the interior points. No finite difference approximation is needed for the boundary points. The remaining points, the near boundary points require a special approach since the computational stencil in (172) is not universally applicable. We'll defer discussion of the near-boundary point issue till later. Suppose all values of $u_{k,l}$ are either members of the set of interior or boundary points. Boundary values are known. Interior points are unknown and each is a linear combination defined by (172), i.e. by its nearest neighbors:

(172) can be written as the linear algebraic system of equations

(173) $$A\mathbf{u} = \mathbf{b} \qquad \text{(exercise, show this)}$$

$\mathbf{b}$ includes $(\Delta x)^2 f_{k,l}$ and contributions from boundary values.

So we ask some basic questions about the resulting linear algebraic system:

(1) Is the linear system nonsingular? $\Rightarrow$ finite difference solution $\mathbf{u} \equiv (u_{k,l})_{k,l,interior}$ integers exists and is unique, if so.

(2) Suppose $\mathbf{u}$ exists and is unique. Does $\mathbf{u} \to \mathbf{U}_\Delta$ as $\nabla x \to 0$, and what is the magnitude of the error?

(3) What efficient methods should be used to solve the linear system? This is crucial since likely to be a very large number of equations.

Take $u_{k,l}$ and arrange it into a 1-D vector of size m^2, say. Note that construction of (173) is not uniquely structured: there are $(m^2)!$ ways to arrange it.

Lemma

The matrix A in (172) is symmetric and the set of its eigenvalues is

$$\sigma(A) = \{\lambda_{\alpha,\beta} : \alpha, \beta = 1, 2, \ldots m\}$$

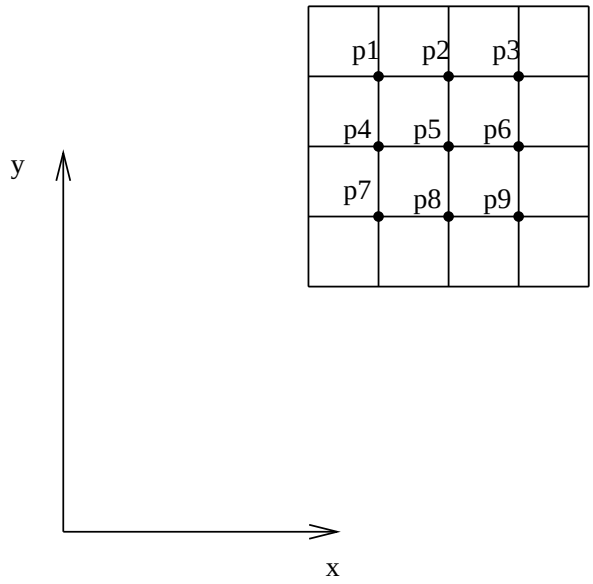

Figure 33: Unit rectangle, gridded with $\Delta x = 1$ and $m = 4$. Points labeled lexicographically.

where

$$\lambda_{\alpha,\beta} = -4\left\{\sin^2\left[\frac{\alpha\pi}{2(m+1)}\right] + \sin^2\left[\frac{\beta\pi}{2(m+1)}\right]\right\} \qquad \text{where} \qquad \alpha, \beta = 1, 2, \cdots m.$$

Proof: One can prove symmetry by examinig the elements of the matrix in a general case. For us it would be more instructive to consider an example and from it see the symmetry: take $\nabla^2 U = 0$ and use stencil with $m = 3$. We'll ignore specific boundaries. Take a domain that's a rectangle and let $\Delta x = 1$. The 5-point approximation is then

$$u_{k-1,l} + u_{k+1,l} + u_{k,l-1} + u_{k,l+1} - 4u_{k,l} = 0.$$

Label elements "lexicographically", as in the Figure 33

Let $w_i \equiv u(P_i)$ so that $w_i = u_{k,l}$ and $P_i \equiv (x_k, y_l)$ where $k = 1, 2, \ldots n$ and $l = 1, 2 \ldots m$. Finally, set $i = k + (m - l)n$ so that the lexicographic label i is consistent with the position label k, l. Then, at each node the equations

are:

$$
\begin{aligned}
P_1: &\quad 4w_1 - w_2 - w_4 = u_{0,3} + u_{1,4} \\
P_2: &\quad 4w_2 - w_3 - w_1 - w_5 = u_{2,4} \\
P_3: &\quad 4w_3 - w_2 - w_6 - = u_{4,3} + u_{3,4} \\
P_4: &\quad 4w_4 - w_5 - w_1 - w_2 = u_{0,2} \\
P_5: &\quad 4w_5 - w_6 - w_4 - w_2 - w_8 = 0 \\
P_6: &\quad 4w_6 - w_3 - w_3 - w_9 = u_{4,2} \\
P_7: &\quad 4w_7 - w_8 - w_4 = u_{0,1} + u_{1,0} \\
P_8: &\quad 4w_8 - w_9 - w_7 - w_5 = u_{2,0} \\
P_9: &\quad 4w_9 - w_8 - w_6 = u_{3,0} + u_{4,1}
\end{aligned}
$$

Why go through this trouble?? This generates a BANDED MATRIX Of bandwidth $= 2n$!! $\therefore$ MUST USE THIS ORDERING FOR LARGE SYSTEMS. As an exercise, construct the matrix problem using any other arrangement and compare the computational characteristics of the resulting matrix against the one we just worked out. So the matrix A is

$$
\left\{
\begin{array}{ccccccccc}
4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4
\end{array}
\right\}
$$

and the resulting system to solve is

$$A\mathbf{w} = \mathbf{b}$$

and $\mathbf{b}$ contains f contributions as well as boundary contributions. So we see it's symmetric. One can show this is a general characteristic of the lexicographic arrangement for this PDE and this computational stencil.

□

Eigenvalues (general case): eigenvalues of A are independent of how A is formed $\rightarrow$ symmetric perturbations conserve eigenvalues.

The eigenvalues problem, in terms of the original values, is

$$u_{k-1,l} + u_{k+1,l} + u_{k,l-1} + u_{k,l+1} - 4u_{k,l} = \lambda u_{k,l} \quad k,l = 1,2\ldots m \tag{174}$$

where λ is an eigenvalue. This is a homogeneous equation, which is further constrained to have eigenfunctions such that $u_{k,0} = u_{k,m+1} = u_{0,1} = u_{m+1,l} = 0$ by the Dirichlet boundary conditions.

Given $\alpha, \beta \in \{1, 2, \ldots m\}$ we have eigenfunctions

$$u_{k,l} = \sin\left(\frac{k\alpha\pi}{m+1}\right)\sin\left(\frac{l\beta\pi}{m+1}\right) \quad k,l = 0,1\ldots m+1$$

which automatically satisfy boundary conditions.

Why this form? Check PDE references on harmonic functions and their connection to the equation $\nabla^2 U = f$.

Substituting $u_{k,l}$ into (174) and exploiting identity $\sin(\phi-\psi)+\sin(\phi+\psi) = 2\sin\phi$ we obtain $\lambda = \lambda_{\alpha,\beta}$

Corollary: The matrix A is negative definite and, a fortiori, nonsingular.

Proof: Already established that A is symmetric. Previous lemma showed that eigenvalues are negative and distinct $\Rightarrow$ nonsingular.

(Recall that all eigenvalues of a symmetric matrix are real, all eigenvalues of a skew-symmetric matrix are purely imaginary, all eigenvalues of a general real matrix are either real or form complex-conjugate pairs. Also, if all eigenvalues of symmetric matrix $> 0 \Rightarrow$ matrix is positive definite. If all eigenvalues of symmetric matrix $< 0 \Rightarrow$ matrix is negative definite.)

□

Remarks:

- $U = \sin(\alpha\pi x)\sin(\beta\pi y)$ is the general solution of $\nabla U = \lambda U$ on $\Omega =$ a unit square in x and y, with $U = 0$ on $\partial\Omega$. α, β are positive integers and $\lambda = -(\alpha^2+\beta^2)\pi$. If we sample U on a grid of points $\left\{\frac{k}{m+1}, \frac{l}{m+1}\right\}$ for $\alpha, \beta = 1, 2, \cdots m$, we obtain the five-point discretization formula for ∇^2 on $\Omega =$ unit square with $(\mathcal{O})$ boundary condition on $\partial\Omega$ and is finite dimensional, of course.

- $(\Delta x)^{-2}\lambda_{\alpha,\beta}$ is a good approximation of $-(\alpha^2+\beta^2)$ is provided α, β small compared m: expanding $\sin^2\theta$ in a series and using $(m+1)\Delta x = 1$

$$\frac{\lambda_{\alpha,\beta}}{(\Delta x)^2} = -4\left(\left\{\left[\frac{\alpha\pi}{2(m+1)}\right]^4 + \cdots\right\} + \left\{\left[\frac{\beta\pi}{2(m+1)}\right]^4 + \cdots\right\}\right)$$
$$= -(\alpha^2+\beta^2)\pi + \frac{1}{12}(\alpha^4+\beta^4)\pi^4(\Delta x)^2 + \mathcal{O}((\Delta x)^4)$$

Theorem (Approximation error):

let $e_{k,l} = U_{\Delta x} - u_{k,l} \quad k,l = 0,1,\ldots m+1$
let $\mathbf{e}$ denote the vector after lexicographic rearrangement
in which approx is posed in terms of $\mathbf{u}$.

Subject to f being sufficiently smooth, $\exists c > 0$ a number independent of Δx such that

$$||\mathbf{e}|| \leq c(\Delta x)^2 \quad \Delta x \to 0$$

where $||\cdot||$ is the Euclidean norm $\left(i.e. ||x|| = [\langle \mathbf{x}, \mathbf{x}\rangle]^{\frac{1}{2}}\right)$ or l_2 norm.

Proof: Homework exercise (hint: since A is symmetric, the l_2 norm = spectral radius).

□

Near-Boundary Points:

Previous analysis works on rectangular domains, L- shaped domains, etc, provided ratios of all sides are rational numbers. In general, however, we expect near-boundary points in which the 5-point formula cannot be implemented. Without loss of generality suppose we seek ∇^2 approximation at point P in Figure 34

Let's ignore y-dependence for now. Given some $z(x)$, approximate z'' at $P \sim x_0$ as linear combination of the values of z at $P, Q \sim x_0 - \Delta_x, T \sim x_0 + \tau\Delta x$. Expanding z about x_0 in Taylor series,

$$\frac{1}{(\Delta x)^2}\left[\frac{2}{\tau+1}z(x_0-\Delta x) - \frac{2}{\tau}z(x_0) + \frac{2}{\tau(\tau+1)}x(x_0+\tau\Delta x)\right]$$
$$= z''(x_0) + \frac{1}{2}(\tau-1)z''(x_0)\Delta x + \mathcal{O}((\Delta x)^2)$$

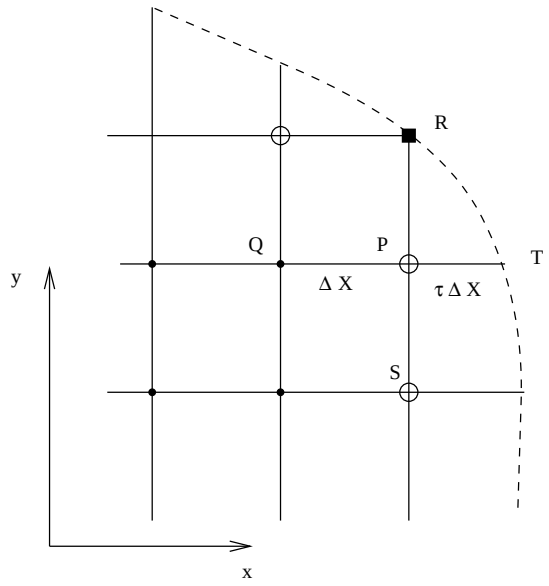

Figure 34: Graphical construction of data at near-boundary point.

unless $\tau = 1$, where everything reduces to central differences, error is just $\mathcal{O}(\Delta x)$. To get $\mathcal{O}(\Delta x^2)$ error, we add the function value at $V \approx x_0 - 2\Delta x$ to linear combination, so that now

$$z''(x_0) = \frac{1}{(\Delta x)^2}\left[\frac{\tau-1}{\tau+2}z(x_0-2\Delta x) - \frac{2(2-\tau)}{\tau+1}z(x_0-\Delta x) - \frac{3-\tau}{\tau}z(x_0) + \frac{6}{\tau(t+1)(\tau+2)}z(x_0+\Delta x)\right] + \mathcal{O}(\Delta x)^2).$$

A good approximation to $\Delta^2 U$ at P $\therefore$ involves 6 points: P, Q, R, S, T, V. Assuming $P = (k^0, l^0)$ in our coordinate system, we get the linear combination

$$\left[\begin{array}{l} \frac{\tau-1}{\tau+2}u_{k^0-2,l^0} + \frac{2(2-\tau)}{\tau+1}u_{k^\circ-1,l^\circ} + \frac{6}{\tau(\tau+1)(\tau+2)}u_{k^\circ+\tau,l^\circ} + u_{k^\circ,l^0-1} \\ +u_{k^\circ,l^\circ+1} - \frac{3+\tau}{\tau}u_{k^\circ,l^\circ} = \Delta x^2 f_{k^\circ,l^\circ} \end{array}\right]$$

where $u_{k^\circ+\tau,l^\circ}$ is the value of U at T, given by boundary conditions.

Note: if $\tau = 1$, we get 5-point formula and P is an internal point, as it should be.

A similar treatment applies in the y-direction ... note that Δx should be small for $\mathcal{O}(\Delta x^2)$ to be small.

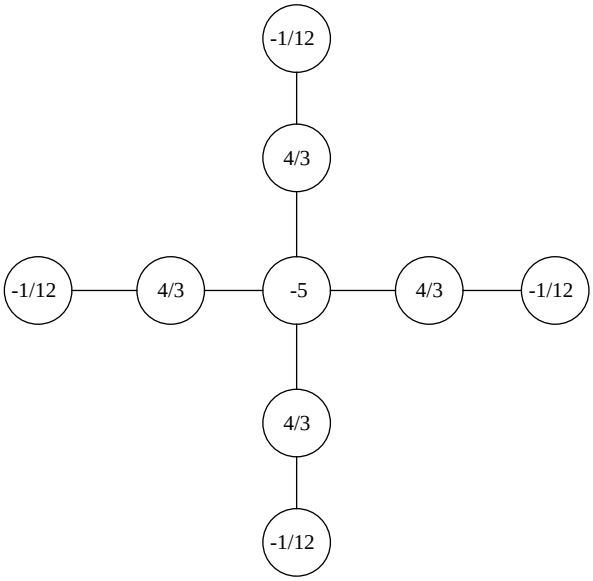

Figure 35: Computational Stencil for the differential operator, using simple 9-point formula.

Higher-Order Finite Difference Methods

Since the 5-point formula gives a $\mathcal{O}(\Delta x^2)$ method, one wonders if it is worth going to higher order: take $\nabla^2 U = f$ and truncate the finite difference approximation using center differences at the next order:

$$\frac{1}{(\Delta x)^2}\left[\delta_x^2 + \delta_y^2 - \frac{1}{12}(\delta_x^4 + \delta_y^4)\right] u_{k,l} = f_{k,l}$$

The resulting computational cell for the differential operator will be as depicted in Figure 35.

Although error $\mathcal{O}(\Delta x^4)$, this is not popular method: renders too many points as near-boundary ones (even on square grid!) requiring laborious treatment. More problematic, however, it gives systems that are considerably more expensive to solve!!

ALTERNATIVELY: The "Nine-Point Formula":

$$\frac{1}{(\Delta)x^2}\left(\delta_x^2 + \delta_y^2 + \frac{1}{6}\delta_x^2\delta_y^2\right) u_{k,l} = f_{k,l}$$

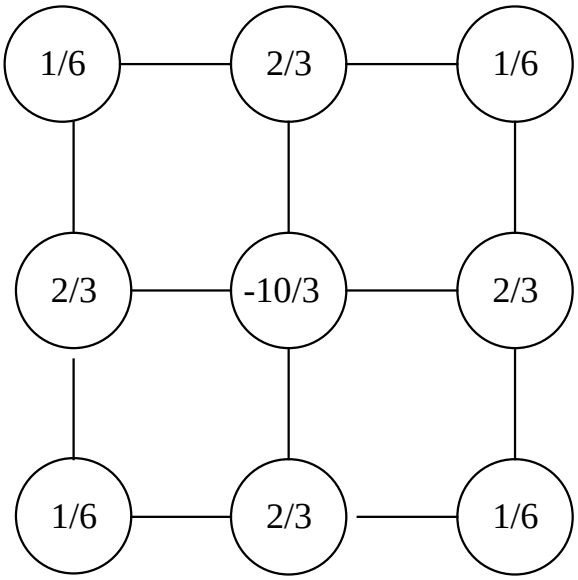

Figure 36: Computational Stencil for the differential operator, using proper 9-point formula.

Computational Cell is given in Figure 36 In homework you will derive the truncation error to be

$$\frac{1}{\Delta x^2}(\delta_x^2+\delta_y^2+\frac{1}{6}\delta_x^2\delta_y^2)\phi-\nabla^2\phi=\frac{1}{12}(\Delta x^2)\nabla^4\phi+\mathcal{O}(\Delta x^2) \tag{175}$$

i.e. same as 5-point!! So apparently, nothing is gained by adding 4 other nearest neighbor points.

In homework you will see that computation with 5-point formula is consistent with error decaying proportionally with $\mathcal{O}(\Delta x^2)$. Will find, however, that for 9-point formula error decays $\mathcal{O}(\Delta x^4)$, in spite of above error estimate result.

Why this discrepancy? Because truncation error estimate above was performed on Laplace equation $\nabla^2 U=0$, not Poisson $\nabla^2 U=f$.

From (175) the 9-point formula is an approximation $\mathcal{O}(\Delta x^4)$ of the equation

$$\Big[\nabla^2+\frac{1}{12}(\Delta x)^2\nabla^4\Big]U=f \tag{176}$$

$$\text{let } \mathcal{L}_{\Delta x} \equiv I + \tfrac{1}{12}(\Delta x)^2 \nabla^2$$

Note $\mathcal{L}_{\Delta x}^{-1}$ exists for sufficiently small Δx. Multiply (176) by $\mathcal{L}_{\Delta x}^{-1}$ and let $f = 0 \Rightarrow$ get $\mathcal{O}(\Delta x^4)$ error.

We can exploit this fact as follows: for (176) with $f \neq 0$, multiply both sides by $\mathcal{L}_{\Delta x}^{-1}$ (for Δx sufficiently small)

$$\nabla^2 U = \mathcal{L}_{\Delta x}^{-1} f \tag{177}$$

is a Poisson equation for which the 9-point formula produces $\mathcal{O}(\Delta x^4)$ error. Not the equation we wanted to solve, but we can do the following:

$$\text{let } \tilde{f}(x,y) = \tilde{\mathcal{L}}_{\Delta x}^{-1} f = f(x,y) + \tfrac{1}{12}\nabla^2 f(x,y) + \mathcal{O}(\Delta x^4)$$

so we see that (177) differs from $\nabla^2 U = f$ by $\mathcal{O}(\Delta x^4)$ terms $\therefore$ it is a good approximation to orignal problem!

$$\left[\begin{array}{l} \text{So the "modified 9-point scheme" is} \\ \\ \frac{1}{\Delta x^2}\left(\delta_x^2 + \delta_y^2 + \frac{1}{6}\delta_x^2\delta_y^2\right) u_{k,l} = [I + \frac{1}{12}(\delta_x^2 + \delta_y^2)] f_{k,l} \\ \\ \text{is } \mathcal{O}(\Delta x^4) \\ \\ \end{array}\right]$$

This is an example of "Preconditioning". It is a very powerful way to get either a better truncation or a faster solution to linear systems with extra computation that is minimal... See homework.

SOLVING THE RESULTING SYSTEM

Define the resulting system

$$A\mathbf{u} = \mathbf{b} \tag{178}$$

Let's look at the structure of A and from there decide what are some methods for solving the linear system. For both the 5-point as well as the 9-point formula it is clear that the matrices are symmetric and diagonally dominant. For small sized problems, direct solution is fine. For large problems, the fastest algorithm would have at its core a conjugate gradient iterative solve.

Let's consider the 5-point formula in some detail. Schematically, the matrix can be expressed in terms of smaller matrices:

$$A = \begin{bmatrix} C & I & 0 & & 0 \\ I & C & I & & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ : & & I & C & I \\ 0 & & 0 & I & C \end{bmatrix}$$

Where C is tridiagonal and I is the identity matrix, where

$$I = \begin{bmatrix} 1 & & & & \\ & 1 & & 0 & \\ & & \ddots & & \\ & 0 & & 1 & \\ & & & & 1 \end{bmatrix} \text{ and } C = \begin{bmatrix} -4 & 1 & 0 & \vdots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ : & \ddots & 1 & -4 & 1 \\ 0 & \ddots & 0 & 1 & 4 \end{bmatrix}$$

diags are all equal

A is said to be TST (Toeplitz Symmetric Tridiagonal), and each block is itself TST.

See class notes from 475A for a review of solving (178). Direct Method: OK for small matrices. Use a Cholesky factorization $A = LL^T$. Such factorization will take a time T which is roughly proportional to number to nonzero elements in main diagonal. For sparse matrices, this is not too big a deal in terms of storage.

Iterative Methods

Incomplete $LU \rightarrow$ splits A into the tridiagonal part and the rest. Iteration is carried out on the tridiagonal part. Also called "line relaxation method." The method is of low cost and is easy, but not the fastest.

ILU FACTORIZATION (Incomplete LU)

Suppose you want to solve

(179) $$A\mathbf{v} = \mathbf{b} \quad v \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^n \quad A \text{ matrix is } n \times n$$

Recall:

A linear one-step stationary scheme would lead to

$$\mathbf{x}^{k+1} = T\mathbf{x}^k + \mathbf{c} \quad \mathbf{x} \in \mathbb{R}^n \tag{180}$$

k is iteration index.

such that $\lim_{k\to\infty} \mathbf{x^{k+1}} = \mathbf{v}$

and $\mathbf{x}^{k+1} = t_k(\mathbf{x^0}, \mathbf{x^1}, \mathbf{x^2} \cdots \mathbf{x^k}) \quad k = 0, 1, 2, \cdots$

$$t_k = \underbrace{\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^\kappa}_{k} \to \mathbb{R}^\kappa \quad k = 0, 1, 2 \cdots \tag{181}$$

Let's go back to (179): suppose that A is in the form $A = \tilde{A} - E$ where the underlying LU factorization of $\tilde{A}$ (nonsingular) can be done easily. For example $\tilde{A}$ may be bounded (or in the context of elliptic problems, TST). Moreover assume E is small compared to $\tilde{A}$. Rewrite (179) as:

$$\tilde{A}\mathbf{v} = E\mathbf{v} + \mathbf{b}$$

which suggests the iterative scheme

$$\tilde{A}\mathbf{x}^{k+1} = E\mathbf{x}^k + \mathbf{b} \tag{182}$$

This is the ILU (incomplete LU factorization). It requires a single LU (or Cholesky if $\tilde{A}$ symmetric) factorization, which can be reused at each iteration.

We can write (182) in the form of (180): let $T = -\tilde{A}^{-1}E$ and $\mathbf{c} = \tilde{A}^{-1}\mathbf{b}$

$$\begin{aligned} \therefore (I - T)A^{-1}\mathbf{b} &= (I - \tilde{A}^{-1}E)(\tilde{A} - E)\mathbf{b} = \tilde{A}^{-1}(\tilde{A} - E)(\tilde{A} - E)^{-1}\mathbf{b} \\ &= \tilde{A}^{-1}\mathbf{b} = \mathbf{c} \end{aligned}$$

Of course, numerically, we solve the form given by (182) and not in the form directly above.

□

The ILU iteration (182) is an example of "regular splitting." More generally $A = P - N$, where P is nonsingular matrix, and build

$$P\mathbf{x}^{k+1} = N\mathbf{x}^k + \mathbf{b} \quad k = 0, 1, 2 \cdots \tag{183}$$

Where P is simple to solve (using LU or other means). Note that, formally, $T = P^{-1}N = P^{-1}(P - A) = I - P^{-1}A$ and $\mathbf{c} = P^{-1}\mathbf{b}$.

Theorem: Suppose both A and $P + P^T - A$ are symmetric and positive definite. Then (183) converges.

Proof: See numerical linear algebra book. □

Remarks What other techniques can we use? From the classical iterative techniques we could use use Jacobi, Gauss-Seidel, SOR, Conjugate Gradient. SOR would be a good choice since we can tune the relaxation parameter and get speeds that exceed Jacobi and Gauss-Seidel, since we can easily compute the spectral radius of the matrix problem that results from the 5-point scheme. Conjugate Gradient is the other logical choice. Later on we'll introduce another fast solver technique, call "Multi-grid" techniques. Are there faster ways? One of the most important developments in linear algebra in recent years has not been in the area of new methods for the solution (although plenty of new idea and schemes have been developed) but where significant strides are made is in algorithmic aspects of the solver techniques: computer-science solutions to increase the speed or efficiency in storage of general linear algebra solvers. See Demmel's linear algebra book for full details on the latest techniques.

FASTER ITERATIVE TECHNIQUES FOR POISSON PROBLEM

One method that clearly out performs SOR is ADI: let $A = A_x + A_y$ where A_x originates in the x-direction central difference and A_y originates in y-direction central differences.

Assuming that $\tilde{A}\mathbf{x^{k+1}} = E\mathbf{x^k} + \mathbf{b} \quad k = 0, 1, 2, \ldots$ represents an ILU decomposition to the problem $Ax = b$, we can write

$$(A_y - 2I)x^{k+1} = -(A_x + 2I)x^k + \mathbf{b} \quad k = 0, \lim$$

column-wise line relaxation

$$(A_x - 2I)x^{k+1} = (A_y + 2I)x^k + \mathbf{b} \quad k = 0, 1, 2, \cdots$$

row-wise line relaxation.

So generally, choose parameters $\alpha_0, \alpha_1, \cdots$ and iterate

$$\text{ADI} \begin{cases} (A_x - \alpha_{2k}I)x^{2k+1} = -(A_y + \alpha_{2k})x^{2k} + \mathbf{b} \\ (A_y - \alpha_{2k+1}I)x^{2k+2} = -(A_y + \alpha_{2k+1})x^{2k+1} + \mathbf{b} \quad k = 0, 1, \cdots \end{cases}$$

The choice of $\{\alpha_k\}$ that beats SOR is given in

[Wachpress (1966) "Iterative Solution of Elliptic Equations . . ."
Prentice-Hall. Also, see Demmel's Linear Algebra book.

Another method (which is highly recommended) is the Conjugate gradient method (see details in Math 475A notes) which is applicable to symmetric positive definite A's. It is very fast.

The conjugate gradient and the preconditioned congugate gradient methods are widely available as mature code (see Netlib). These methods are part of a family of methods called "Krylov Space Methods." Other Krylov-space methods worth knowing about are (GMRes) Generalized Minimal Residual and bi-conjugate gradient methods.

There are other types of methods for the solution of the Poisson equation and some of its close cousins: there's "cyclic reduction and factorization," there is the very fast "Multipole Techniques." We will feature two more methods here, both are very powerful and are widely applicable. First we'll consider "FFT-Based Poisson Solvers" and then briefly talk about "Multigrid Methods."

Remarks As a general rule of thumb, if the problem you're solving is very large, you'll have to resort to high performance solvers. But the first thing you should do is decide whether you have a problem that's big enough to warrant looking at high performance algorithms...this includes foreseeing that in the future you might look at big problems. Alternatively, suppose you have to solve the same problem millions of times (well, many times), whether it is small or large, you could save yourself time and storage by using a high performance solver. In any event, it doesn't hurt to be aware that these sort of things exist and most likely are in the form of mature code that you can adapt to your problem.

"FAST" POISSON SOLVER

These are fast, in the sense that if you use an FFT to do the Fourier computations, it is faster that the Discrete Fourier Transform (DFT). Competitive in speed with ADI and SOR optimized. It proves useful, in any event, to know how to use this technology because its applicable to other boundary

value problems.

Motivations There are a number of reasons for visiting this problem:

- You get an idea of what is involved in solving problems in two-space dimensions and of the importance of boundary data in fixing a solution.
- You see how advantageous it is, in many cases, to use analytical means as much as possible to pose a problem for solution BEFORE actually coding it up. In general, one should explore all possible means to advance a calculation by analytical means before resorting to numerics...of course, this is not a theorem, but merely a rule of thumb.
- Following the lead in the last item, we could find that the problem posed below can be solved exactly via analytical means (as are many of the problems covered in this course). Nevertheless, we want to remind you that in many instances, problems in several space dimensions can be solved most easily numerically and analytically if you happen to choose the right reference frame and/or coordinate system. In this instance we'll emphasize the issue of choosing the coordinate system, and in this case, the choice is based on symmetries in the boundary geometry and the type of PDE (consult an elementary PDE book, particularly, one geared towards engineers).
- The reason for wanting to solve the disk problem is that we'll get come practice in solving PDE in coordinates other than Cartesian, and we'll show how the boundary conditions must be payed special attentions to.
- We will also use this problem to introduce, albeit in an elementary way, how spectral methods can be used numerically. Also, many books do not emphasize the fact that numerical methods can often be combined, a fact that is obvious later on, but sometimes not so obvious as a beginning computational scientist.

We will solve the Helmholtz Equation

$$\left(\nabla^2 + \kappa^2\right) U = g \begin{cases} g, & \text{real} \\ k, & \text{real} \\ u, & \text{real} \end{cases} \tag{184}$$

on a unit disk with domain $D = \left\{(x,y) \in R^2 : x^2 + y^2 < 1\right\}$

with boundary conditions $\begin{cases} U(\cos\theta, \sin\theta) = \phi(\theta) & 0 \le \theta \le 2\pi \\ \phi(0) = \phi(2\pi) \quad , & \text{Dirichlet boundary conditions} \end{cases}$

Remarks Note that (184), with $k^2 = 0$, we get the Poisson Equation. The Helmholtz Equation originates from the linear 2-way wave equation, where the time dependence of the solution is assured time harmonic:

To see this, take the Wave Equation (with no forcing term, for simplicity)

$$\frac{1}{c^2}\psi_{tt} - \nabla^2\psi = 0 \quad c^2 > 0 \text{ constant}, \tag{185}$$

where c is the wave speed.

Substitute

$$U(x,y) = \psi e^{-i\omega t} \tag{186}$$

where ω is the frequency of the wave, assumed constant. Let $\dfrac{\omega}{c} = \kappa$ be the wavenumber, then the (186) solution to (185) is found by solving

$$(\kappa^2 + \nabla^2)U = 0 \tag{187}$$

When the problem is defined on a disk and a forcing term g is added to the equation (187), we get (184).

Computational Grid

The symmetry of the problem leads us to try a change of coordinate system, from Cartesian to polar: The mapping: $\begin{cases} x = r\cos\theta \\ y = r\sin\theta \\ x^2 + y^2 = r^2 \end{cases}$

Hence, in the polar coordinate system we obtain a square lattice $\widetilde{\mathcal{D}}$ associated with $\mathcal{D}$, the cylindrical domain via the mapping above. The square grid $\widetilde{\mathcal{D}}$ has four edges, denoted by $\partial\widehat{\mathcal{D}}$. The two domains are illustrated in Figure 37.

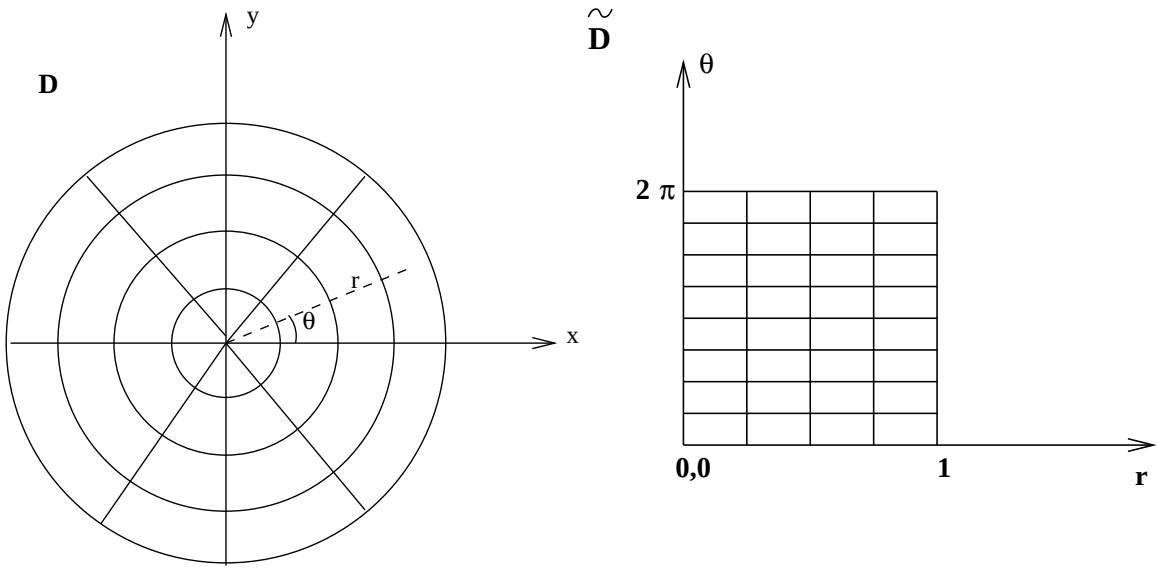

Figure 37: The Cartesian and Polar domains $\mathcal{D}$ and $\tilde{\mathcal{D}}$, respectively

$$\text{let} \quad U(r,\theta) = U(r\cos\theta, r\sin\theta) \quad \begin{cases} 0 < r < 1 \\ 0 \le \theta \le 2\pi \end{cases}$$
$$g(r,\theta) = g(r\cos\theta, \sin\theta)$$

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}(\text{cartesian}) \Rightarrow \nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial\theta} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} \quad \text{(Polar)}$$

$\therefore$ (184) in Polar Coordinates is:

$$(188) \qquad \therefore \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial\theta^2}u + \kappa^2 u = g \quad \begin{cases} 0 < r < 1 \\ 0 \le \theta \le 2\pi \end{cases}$$

Boundary Conditions: on D we have Dirichlet condition on the edge of the disk. Two other conditions are based on our expectation of the solution. For example, we could demand that the solution be bounded at the center of the disk and that it be periodic in θ.

$\widetilde{D}$ has 4 edges for which we need 4 conditions, that is, on $\partial\widetilde{D}$: $r = 1, 0 \le$

$\theta \leq 2\pi$

$$\begin{aligned} &\text{the } U(r\cos\theta, r\sin\theta) = \phi(\theta) \text{ gives} \\ &\Rightarrow u(1,\theta) = \phi(\theta) \quad 0 \leq \theta \leq 2\pi. \\ \text{on } 0 < r < 1 \quad \theta = 0, \quad &\text{and} \quad 0 < r < 1, \theta = 2\pi \\ u(r,0) = u(r,2\pi) \quad &0 < r < 1 \quad \text{(Periodic)} \end{aligned}$$

Finally, we need a condition at $r = 0$, for $0 \leq \theta \leq 2\pi$. This whole line corresponds to just a single point in D, namely $x = 0, y = 0$: the procedure outlined in problem # 3 HW 9 is a general procedure for determining the condition at the origin. Here we could impose the condition that u be constant along the line $r = 0$. This implies that

$$\frac{\partial u}{\partial \theta}(0,\theta) = 0 \quad 0 \leq \theta \leq 2\pi$$

At this point, we have a well-posed problem. We solve (188) on $\widetilde{D}$ subject to

$$\begin{aligned} &u(1,\theta) = \phi && 0 \leq \theta \leq 2\pi \\ &u(r,0) = u(r,2\pi) && 0 < r < 1 \\ &\frac{\partial u}{\partial \theta}(0,\theta) = 0 && 0 \leq \theta \leq 2\pi \end{aligned}$$

Remark: The coordinate transformation is not only a proper choice of the coordinate system, preserving symmetrices of the solution: it also avoids grid errors due to interpolation which would be required in the discretization. Notice as well, that the coordinate system turned our problem into a "separable" PDE (see elementary PDE book).

We can use finite differences to solve the problem and then use the mapping to go from $\widetilde{\mathcal{D}}$ to $\mathcal{D}$ and thus obtain the approximate solution $u(x,y)$. We will use Fourier methods, combined with finite differences instead. Fourier methods are particularly suited to the solution of L_2 functions which are periodic. In this case, the periodicity is in θ hence in the θ direction we'll use Fourier, and we'll use finite differences for the radial direction The fastest algorithm for a discrete Fourier transform is the FFT (a useful reference for Fourier Methods is "The DFT" by the W. Briggs & V.E. Henson. The reason we call this section "Fast Fourier Solver for the Poisson/Helmholtz Equation" is because we're using FFT's rather than DFT's. In this sense it is fast.

Since the solution is periodic in θ, we'll apply the Fourier transform:

$$\widehat{u}_m(r) = \frac{1}{2\pi}\int_0^{2\pi} u(r,\theta)e^{-im\theta}d\theta \quad m \in \mathbb{Z}$$

$\widehat{u}$ is complex (in the implementation we could use a sine or cosine transform, but we will use the complex transform, for simplicity and generality). The goal is to convert the PDE into an infinite set of ODE's for the Fourier coefficients $\{\hat{u}_m\}_{m=-\infty}^{\infty}$.

It's easy to deduce that

$$\begin{aligned}
\widehat{\left(\frac{\partial u}{\partial r}\right)}_u &= u'_m(r) \qquad \text{and} \widehat{\left(\frac{\partial^2 u}{\partial r^2}\right)}_m = \widehat{u}''_m(r) \quad m \in \mathbb{Z} \\
\widehat{\left(\frac{\partial^2 u}{\partial \theta^2}\right)}_m &= \frac{1}{2\pi}\int_0^{2\pi}\frac{\partial^2 u}{\partial\theta^2}(r,\theta)e^{-im\theta}d\theta = \frac{1}{2\pi}\left[e^{-im\theta}\frac{\partial u(r,\theta)}{\partial\theta}\Big|_0^{2\pi} + im\int_0^{2\pi}\frac{\partial u(r,\theta)}{\partial\theta}e^{-im\theta}d\theta\right] \\
&= \frac{im}{2\pi}\int_0^{2\pi}\frac{\partial u(r,\theta)}{\partial\theta}e^{-im\theta}d\theta = \\
&= \frac{im}{2\pi}\left[e^{-im\theta}u(r,\theta)\Big|_0^{2\pi} + im\int_0^{2\pi}u(r,\theta)e^{-im\theta}d\theta\right] = -m^2\hat{u}_m(r)
\end{aligned}$$

So, we multiply (188) by $e^{-im\theta}$ and integrate from 0 to 2π and divide by 2π. Using orthoqonality, we obtain the set of ODE's for the Fourier coefficients:

$$\widehat{u}''_m + \frac{1}{r}\widehat{u}'_m - \frac{m^2}{r^2}\widehat{u}_m + \kappa^2\widehat{u}_m = \widehat{g}_m \quad \text{for } r \in (0,1) \qquad m \in \mathbb{Z} \tag{189}$$

We also transform the boundary conditions:

$$\widehat{u}_m(1) = \widehat{\phi}_m \quad m \in \mathbb{Z}$$

(Note that solution is already periodic in 2π by having used Fourier transforms). Now we need to take care of the condition $\dfrac{\partial}{\partial\theta}u(0,\theta) = 0$:

$$\text{Since } u(r,\theta) = \sum_{m=-\infty}^{\infty}\widehat{u}_m(r)e^{im\theta} \quad 0 \le r \le 1, \quad 0 \le \theta \le 2\pi$$

differentiate with respect to θ and set $r = 0$:

$$i\sum_{m=-\infty}^{\infty} m\hat{u}_m(0)e^{im\theta} \equiv 0, \quad 0 \le \theta \le 2\pi$$

Since the $e^{im\theta}$ are linearly independent, for $m \in \mathbb{Z} \Rightarrow$ all coefficients must vanish identically:

$$\hat{v}_m(0) = 0 \quad m \in \mathbb{Z}\backslash\{0\}$$

This leaves us with just a single missing item of information, namely the boundary condition for the zeroth harmonic.

$$\text{Since } u(r,\theta) = U(r\cos\theta, r\sin\theta)$$

$$\therefore \frac{\partial u(0,\theta)}{\partial r} = \cos\theta \frac{\partial U(0,0)}{dx} + \sin\theta \frac{\partial U(0,0)}{\partial y}$$

$$\therefore \hat{u}'(0) = \frac{1}{2\pi}\int_0^{2\pi} \frac{\partial u(0,\theta)}{\partial r} d\theta = \frac{1}{2\pi}\int_0^{2\pi} \cos d\theta \frac{\partial U(0,0)}{\partial x} + \frac{1}{2\pi}\int_0^{2\pi} \sin\theta \frac{\partial U(0,0)}{\partial y} = 0$$

Since

$$\int_0^{2\pi} \cos\theta d\theta = \int_0^{2\pi} \sin\theta d\theta = 0 \quad \therefore$$

$$u'(0) = 0 \tag{190}$$

Now, we're ready on the ODE part.

Remark: The key fact is that the fourier transform uncouples the harmonics. We need to solve the ODE system and then use the "inverse" Fourier formula

$$u(r,\theta) = \sum_{m=-\infty}^{\infty} \hat{u}_m(r) e^{im\theta}$$

It turns out that (189) has an analytical solution in terms of Bessel functions. More generally, we would use one of the boundary value techniques presented in this course, i.e. finite differences, shooting method, FEM.

Exercise) Show that, by using center differences, the FD approximation $w(k\Delta r) = u(k\Delta r) + \mathcal{O}(\Delta r^2)$ is given by the solution of

(191)

$$\begin{cases} (1-\frac{1}{2k})\widehat{w}_{m,k-1} - \left(2+\kappa^2+\frac{m^2}{k^2}\right)\widehat{w}_{m,k} + \left(1+\frac{1}{2k}\right)\widehat{w}_{m,k+1} = (\Delta r)^2 \widehat{g}_{m,k} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 1,2,\cdots d-1 \quad , \quad m \in \mathbb{Z} \\ \text{with } \Delta r = \dfrac{1}{d} \\ \text{subject to } \widehat{w}_{m,d} = \widehat{\phi}_m \quad m \in \mathbb{Z} \\ \widehat{w}_{m,0} = 0 \quad m \in \mathbb{Z}\backslash\{0\} \\ -\dfrac{5}{4}\widehat{w}_{0,0} + \dfrac{3}{2}\widehat{w}_{0,1} - \dfrac{1}{4}\widehat{w}_{0,2} = 0 \quad \text{(forward difference approx of (190))} \end{cases}$$

The outcome is a tridiagonal system for every $m \neq 0$ and an almost tridiagonal system for $m = 0$. Such systems can be efficiently solved by sparse LU factorization.

Implementation comments: Of course, we need to truncate the infinite set of ODE's by a subset. Say

$$-M + 1 \leq m \leq M$$

(the truncation error induced depends on how large M is. See Canutto, Quarteroni, Hussaini, book on Spectral Methods). Provided ϕ and g are smooth, good accuracy is attained and the error drops exponentially with M!!

(1) Use an FFT, so pick $M = 2^{n-1} \quad n = 1, 2, 3, \cdots$ transform g and ϕ to get $\widehat{g}$ and $\widehat{\phi}$
Pick d and solve (191) for $\widehat{w}_{m,k}$ for $-M + 1 \leq m \leq M$ and $k = 1, 2, \cdots d - 1$

(2) Then employ a $d - 1$ inverse FFT to produce w on a $d \times (2m)$ square grid.

(3) find approximation of u on D by using the mapping on $\widetilde{D}$.

Roughly, we need $\mathcal{O}(M \log_2 M)$ ops for FFT in (186), $\mathcal{O}(dM)$ for LU solution in FD solution in (186), plus $\mathcal{O}(dM \log_2 M)$ for the reconstruction. This compares very favorably with full finite difference methods.

0.7.2 Fundamentals of Multigrids Methods

Some good references: McCormick's "Multilevel Methods for PDE's" (SIAM) and "Multigrid Tutorial" by W. Briggs (SIAM). These are very accessible and inexpensive SIAM books. Here, we'll follow Iserles rather closely.

A good software and information source for multigrid

Multigrid methods are nested techniques for the iterative solution of the linear algebraic problem

$$A\mathbf{x} = \mathbf{b}$$

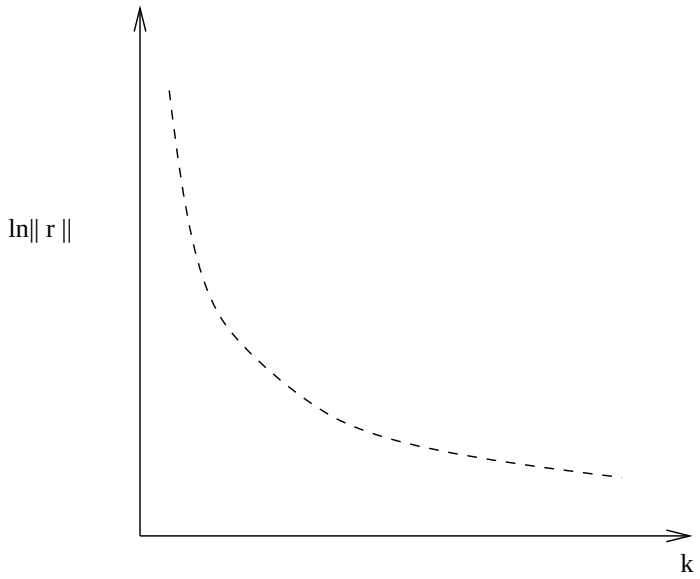

Figure 38: The logarithm of the norm of the residual as a function of the iteration count k for Gauss-Seidel on the Poisson problem

and is a current active research area. Since they are fast, they are used not only in the solution of PDE's but also in applications related to image processing, filtering, etc.

Suppose we want to solve the 5-point FD approximation of the Poisson equation using Gauss-Seidel. As we have discussed in 475A, the rate of convergence of the iterative solution $\rho(A)$, which in this case is approximately $1 - \pi^2/m^2$ for the $m \times m$ matrix A. This is an asymptotic result. In reality, we'd see that $\ln ||r^k||$, where $\mathbf{r}^k = A\mathbf{x}^k - \mathbf{b}$, will drop as shown in Figure 38

we see a severe drop in the first few iterates, followed by the linear rate predicted by the asymptotic result. This is true for any m!!

Why? Because the Gauss Seidel acts as a "smoother," altervating high wave numbers faster than low wave numbers. Understanding why provides a technique for accelerating iterative schemes:

Subtract the 5-point equations

$$u_{j-1,l} + u_{j,l-1} + u_{j+1,l} + u_{j,l+1} - 4u_{j,l} = \Delta x^2 f_{j,l} \quad j,l = 1,2\ldots m \tag{192}$$

from Gauss Seidel Scheme:

$$u^{k+1}_{j-1,l} + u^{k+1}_{j,l-1} + u^{k}_{j+1,l} + u^{k}_{j,l+1} - 4u^{k+1}_{j,l} = \Delta x^2 f_{j,l} \quad j,l = 1,2,\dots m$$

(193) to obtain

$$\varepsilon^{k+1}_{j-1,l} + \varepsilon^{k+1}_{j,l-1} + \varepsilon^{k}_{j+1,l} + \varepsilon^{k}_{j,l+1} - 4\varepsilon^{k+1}_{j,l} = 0 \quad j,l = 1,2\dots m \tag{194}$$

where $\varepsilon^k_{j,l} \equiv u^k_{j,l} - u_{j,l}$ is the error after k iterations at the (j,l) grid point.

Since we're assuming Dirichlet boundary conditions $u_{j,l}$ and $u^k_{j,l}$ are identical at the boundary $\Rightarrow \varepsilon^k_{j,l} = 0$ there.

Let

$$p^k(\theta,\psi) = \sum_{j=1}^{m}\sum_{l=1}^{m} \varepsilon^k_{j,l} e^{i(j\theta + l\psi)}, \qquad 0 \le \theta\psi \le 2\pi$$

be the 2-D Fourier transform of the sequence $\{\varepsilon^k_{j,l}\}^m_{j,l=1}$.

and denote the Euclidean Norm

$$|||g||| = \left[\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |g(\theta,\psi)|^2 d\theta d\psi\right]^{\frac{1}{2}}.$$

By Parseval's Theorem, can show that

$$|||p^k|||^2 = ||\varepsilon^k||^2$$

where $||y|| = \left(\sum_{j=1}^{m}\sum_{l=1}^{m} |y_{j,l}|^2\right)^{\frac{1}{2}}$.

We wish to establish the rate of decay of residuals. Multiply (194) by $e^{i(j\theta+l\psi)}$ and sum over $j,l = 1,2\dots m$

Since

$$\sum_{j=1}^{m}\sum_{k=-1}^{m} \varepsilon_{j-1,l} e^{i(j\theta+l\psi)} = \sum_{j=0}^{m-1}\sum_{l=1}^{m} \varepsilon^{k+1}_{j,l} e^{i[(j+1)\theta+l\psi]} = e^{i\theta} p^k(\theta,\psi)$$

$$-e^{i(m+1)\theta}\sum_{l=1}^{m} \varepsilon_{m,l} e^{il\theta}$$

and applying similar algebra to other term in (194) we obtain

$$\begin{aligned}(4-e^{i\theta}-e^{i\psi})p^{k+1}(\theta,\psi) &= (e^{-i\theta}+e^{-i\psi})p^{k}(\theta,\psi)\\ &- \left\{e^{i(m+1)\theta}\sum_{l=1}^{m}\varepsilon_{m,l}^{k+1}e^{il\psi}+e^{i(m+1)\psi}\sum_{j=1}^{u}\varepsilon_{jm}^{k+1}e^{ij\theta}\right.\\ &+ \left.\sum_{l=1}^{m}\varepsilon_{1,l}^{k}e^{il\psi}+\sum_{j=1}^{m}\varepsilon_{j-1}^{k}e^{ij\theta}\right\}\end{aligned}$$

Now $(4-e^{i\theta}-e^{i\psi})p^{k+1}(\theta,\psi)\approx(e^{-i\theta}+e^{-i\psi})p^{k}(\theta,\psi)$ in general but the term in curly brackets would have disappeared if we were considering periodic boundary conditions. For Dirichlet, the justification is more subtle. We could check a posteriori that it is indeed ok.

Define the *local attenuation factor* as

$$\rho^{k}(\theta,\psi)=\Big|\frac{p^{k+1}(\theta,\psi)}{p^{k}(\theta,\psi)}$$

for $\theta|\leq\pi$ and $|\psi|\leq\pi$,

then $\rho^{k}(\theta,\psi)\approx\overline{\rho}(\theta,\psi)=\left|\frac{e^{-i\theta}+e^{-i\psi}}{4-e^{i\theta}-e^{i\psi}}\right| \qquad |\theta|,|\psi|\leq\pi.$

$\bar{\rho}$ is independent of k.

In HW10 you will confirm graphically that when $\tilde{\rho}$ is restricted to the set $\Pi_0\equiv\left\{(\theta,\psi):\frac{1}{2}\pi\leq\max\left\{|\theta|,|\psi|\right\}\leq\pi\right\}$ the function $\tilde{\rho}$ peaks at $\dfrac{1}{2}$, whereas $\tilde{\rho}$ over $\Pi\equiv\{(\theta,\psi):|\theta|\leq\pi,|\psi|\leq\pi\}$ peaks at 1. In fact, you will show that

$$\max_{(\theta,\psi)\in\Pi}\tilde{\rho}(\theta,\psi)=\tilde{\rho}\left(\frac{\pi}{2},\tan^{-1}\left(\frac{3}{4}\right)\right)=\frac{1}{2}$$

$\therefore$ if we disregard non-oscillatory wave numbers, the amplitude of the error is halved in each iteration!

$\therefore$ G-S attenuates highly oscillatory components much faster thus contribution of these vanishes quickly after a few iterations.

In the context of a continuum, all wavenumbers are supported. On a discretization of the continuum, a lattice or grid what is "highly oscillatory

depends on the grid spacing. In fact, for a specific grid there are oscillations that are not resolvable. Since what is meant by "high oscillations" is with respect to each grid realization, that is, high oscillations are those components with wavelengths that are comparable to the grid size, and the G-S iteration attenuates rapidly "high oscillations" we could conceive of an algorithm in which we go back and forth between coarse and fine meshes and iterate at each level ONLY while attenuation rates are high, we can get a fast rate of convergence for the residual by constructing a nesting sequence between coarse and fine meshes.

Suppose that we coarsen a grid by taking out every second point, the outcome being a grid on $[0,1] \times [0,1]$ but with Δx replaced by $2\Delta x$. The range of former high frequencies Π_0 is no longer visible on the coarse grid. The new grid will have its own range of high frequencies on which G-S performs well:

$$\Pi_1 = \left\{ (\theta, \psi) : \frac{1}{4}\pi \leq \max\{|\theta|, |\psi|\} \leq \frac{1}{2}\pi \right\}$$

We could coarsen again and again and form a hierarchy of grids embedded into each other, whose (grid-specific) high frequencies correspond, as far as the fine grid is concerned, to the sets

$$\Pi_s = \left\{ (\theta, \psi) : 2^{-s-1}\pi \leq \max\{|\theta|\psi|\} \leq 2^{-s}\pi \right\}$$
$$s = 1, 2, 3 \cdots \log_2(m+1)$$

The sets Π_s nest inside each other, as shown in Figure 39

The multigrid technique takes advantage of this fact, traveling up and down the grid hierarchy, using G-S iterations to dampen the locally highly oscillatory components of the error (see Figure 40). We need to describe how each coarsening or refinement step is performed, as well as to specify the exact strategy of how to start, when to coarsen, when to refine and when to terminate the whole process:

Refinement and Coarsening: Consider just 2 grids, find and coarse. Suppose we wish to solve

$$A_f \mathbf{x}_f = \mathbf{v}_f \qquad \text{on a fine grid}$$

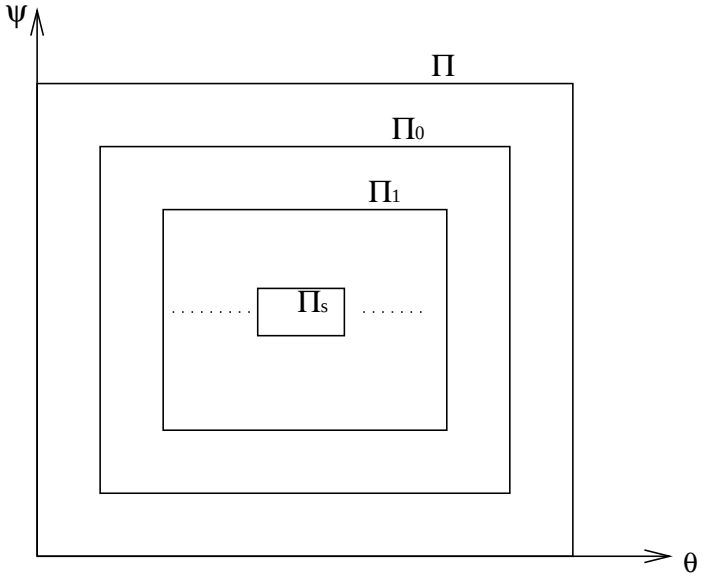

Figure 39: The Nested subspaces

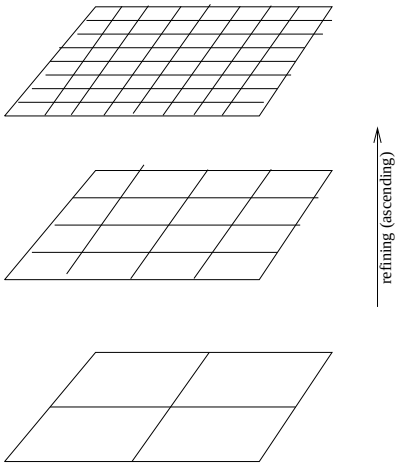

Figure 40: Schematic representation of the grid hierarchy

perform a few G-S iteratives (smoothing out high frequencies), then

$$\mathbf{r}_f \equiv A_f \mathbf{x}_f - \mathbf{v}_f \text{ is the residual.}$$

To go to coarse grid, use a "restriction matrix" R

$$\mathbf{r}_c = R\mathbf{r}_f$$

remember, at this point $\mathbf{r}_f$ is constructed of low frequency components (relative to fine grid) $\therefore$ makes sense to go on smoothing $\mathbf{r}_c$ on coarser grid: let $\mathbf{v}_c \equiv -\mathbf{r}_c$ and solve

$$A_c \mathbf{x}_c = -\mathbf{r}_c \tag{195}$$

A_c is the coarse matrix, i.e. A restricted to coarse grid.

To ascend, suppose $\mathbf{x}_c$ is approximate solution to (195) after a few iterations we translate $\mathbf{x}_c$ into the fine grid using a "Prolongation matrix" P?

$$\mathbf{y}_f = P\mathbf{x}_c \tag{196}$$

and update the old value of $\mathbf{x}_f$:

$$\mathbf{x}_f^{\text{new}} = \mathbf{x}_f^{\text{old}} + \mathbf{y}_f \tag{197}$$

Evaluating the residual $\mathbf{r}_f^{\text{new}}$, under the assumption that $\mathbf{x_c}$ is exact solution of (195). Since

$$\mathbf{r}_f^{\text{new}} = A_f \mathbf{x}_f^{\text{new}} - \mathbf{v}_f = A_f(\mathbf{x}_f^{\text{old}} + \mathbf{y_f}) - \mathbf{v}_f$$

then using (196) and (197):

$$\begin{aligned}
&\mathbf{r_f}^{\text{new}} = \mathbf{r_f}^{\text{old}} + A_f \mathbf{y}_f = \mathbf{r}_f^{\text{old}} + A_f P \mathbf{x}_c \\
&\therefore \mathbf{r}_f^{\text{new}} = \mathbf{r}_f^{\text{old}} - A_f P A_c^{-1} \mathbf{r}_c. \\
&\text{Since } \mathbf{r}_c = R\mathbf{r}_f \\
&\mathbf{r}_f^{\text{new}} = (\mathbf{I} - A_f P A_c^{-1} R)\mathbf{r_f}^{\text{old}}
\end{aligned}$$

$\therefore$ the sole contribution to the new residual comes from replacing a fine grid by a coarser one. *similar reasoning* is valid even if $\mathbf{x}_c$ is anapproximate solution of course grid problem provided high freq's have been smoothed out.

Now we need to specify the Restriction and Prolongation Matrices:

A popular $\{$ Restriction and Prolongation matrix comes from "Full Weighting" :

It leads to $R = \frac{1}{4}P^T$, which is very convenient.

let

$$\begin{aligned} \mathbf{w}_f &= P\mathbf{w}c \\ \mathbf{w}_c &= \left(w^c_{j,l}\right)^m_{j,l=1} \\ \mathbf{w}_f &= \left(w^f_{j,l}\right)^{2m+1}_{j,l=1} \end{aligned}$$

Full Weighting:

$$w^c_{j,l} = \frac{1}{4}w^f_{2j,2l} + \frac{1}{8}\left(w^f_{2j-1,2l} + w^f_{2j,2l-1^t} + w^f_{2j+1,2l} + w^f_{2j,2l+1}\right)$$
$$+\frac{1}{16}\left(w^f_{2j-1,2l-1} + w^f_{2j+1,2l-1} + w^f_{2j-1,2l+1} + w^f_{2j+1,2l+1}\right) \quad j,l = 1,2,\ldots m$$

Prolongation: use linear interpolation:

$$w^f_{2j-1,2l-1} = w^c_{j,l} \quad j,l = 1,2,\ldots m$$
$$w^f_{2j-2,2l} = \frac{1}{2}\left(w^c_{j,l} + w^c_{j,l+1}\right) \quad j = 1,2,\ldots m-1; l-1,2\ldots m$$
$$w^f_{2j,2l-1} = \frac{1}{2}(w^c_{j,l} + w^c_{j+1,l}) \quad j = 1,2\cdots m; l = 1,2\cdots m-1$$
$$w^f_{2j-2l} = \frac{1}{4}\left(w^c_{j,l} + w^c_{j,l+1} + w^c_{j+1,l} + w^c_{j+1,l+1}\right) \quad j,l = 1,2\ldots m-1$$

$wf = 0$ at boundary ... recall we're dealing with residuals.

□

ALGORITHM V-CYCLE (One of many, the simplest and most popular) The algorithm is shown schematically in Figure 41.

Start and end on finest grid. To start, stipulate inital guess $\mathbf{v_f} = \mathbf{b}$ (original right-hand-side of system). Iterate Gauss-Seidel n_r times.

Evaluate $\mathbf{r}_f$

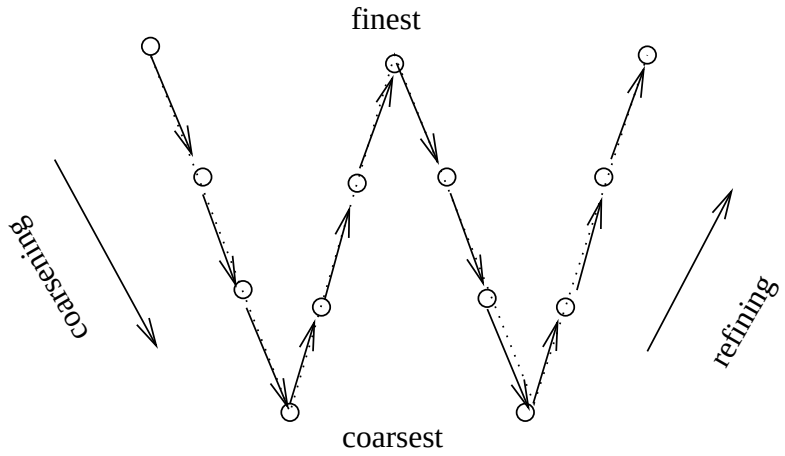

Figure 41: The V-Cycle algorithm

Restrict it to coarser grid

Perform n_r Gauss-Seidel iterations

Evaluate $\mathbf{r}_c$

Restrict on even coarser grid

and repeat process till we reach coarsest grid, with just one single interior point, which we can solve for exactly.

At this stage we've damped out the high frequencies of error, relative to each grid resolution. $\therefore$ damped influence of error components over entire range of wave numbers supported by the finest grid, except for small error introduced by restriction.

Now we go up all the way to the finest grid. At each step we Prolong,

Update residual on new grid,

and Perform no G-S iterations

to eliminate errors (corresponding to high oscillations for each grid resolution) that might have been introduced by post prolongations.

We're back to the starting point, completed the V-Cycle.

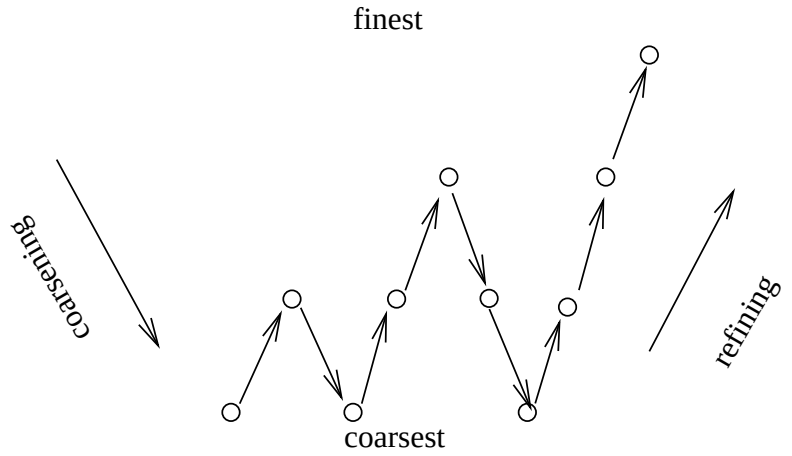

Figure 42: An improved V-cycle algorithm

Now we check for convergence by measuring size of residual vector.

If residual below some specified tolerance, we quit. Otherwise, repeat V-Cycle.

There is one problem with V-Cycle algorithm . . . it could be made faster if we start with a really good initial guess. The "Full Multigrid" method usually combines this with the pattern illustrated in Figure 42. See references for full details.

□

V-Cycle Computational Cost for Poisson (5-point formula):

let γ be the cost of a single G-S iteration on finest grid. Note that a single coarsening decreases operation count by 4 $\therefore$ the cost of V-cycle is

$$\left(1 + \frac{1}{4} + \frac{1}{4^2} + \frac{1}{4^3} \cdots\right)(n_r + n_p)\gamma \approx \frac{4}{3}(n_r + n_p)\gamma \tag{198}$$

where n_r and n_p are the # of iterates in restriction and prolongation phases. $\therefore$ V-cycle is linear in the # of grid points on finest grid.

Ex) For $m = 63$, using V-cycle with $n_r = np = 1$ for residual $\sim 10^{-5} \rightarrow 8^{th} V$-cycle for same residual: using

SOR	$\rightarrow$ 243 iterations
$G-S$	$\rightarrow$ 6526 iterations
Conjugate Gradient	$\rightarrow$ 179

Note: Cost of restriction and prolongation is not included in (198)

□

0.8 APPENDIX

0.8.1 Computing a Matrix Exponential

Consider the matrix

$$e^{At}$$

here t is a scalar parameter, and A is an $n \times n$ matrix.

Lemma: For A and t as above, the eigenvalues of At are t times the eigenvalues of A.

Proof: let μ be the eigenvalue of A then we show that $\lambda = \mu t$ is an eigenvalue of At:
Since μ is an eigenvalue of A then $\det(A - \mu I) = 0$. Hence $\det(At - \mu I) = \det(t(A - \mu I)) = t^n \det(A - \mu 5) = t^n(0) = 0$ □

Lemma:

$$e^{At} = It + At + \frac{1}{2!}At^2 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!} A^n t^n \quad \leftarrow \text{ not generally useful}$$

for any A. □ Theorem: With A as above. Then

$$e^{At} = \alpha_{n-1}A^{n-1}t^{n-1} + \alpha_{n-2}A^{n-2}t^{n-2} + \cdots \alpha_2 A^2 t^2 + \alpha_1 At + \alpha_0 I$$

$\alpha_i \quad i = 0, 1, \cdots, n-1$ are functions of t which must be determined for each A.

Theorem: A as above. Then

$$r(\lambda) \equiv \alpha_{n-1}\lambda^{n-1} + \alpha_{n-2}\lambda^{n-2} + \cdots \alpha_2 \lambda^2 + \alpha_1 \lambda + \alpha_0$$

then if λ_i is an eigenvalue of At then $e^{\lambda i} = r(\lambda i)$

furthermore if λ_i is eigenvalue of multiplicity k, $k > 2$ then the following equations are true:

$$e^{\lambda i} = \frac{d}{d\lambda} r(\lambda)|_{\lambda=\lambda_i} = \frac{d^2}{d\lambda^2} r(\lambda)|_{\lambda=\lambda_i} = \cdots = \frac{d^{k-1}}{d\lambda^{k-1}} r(\lambda)|_{\lambda=\lambda_i}$$

Example Suppose A is 4 matrix, with eigenvalues 5 and 2, with multiplicity $k = 3$ and $k = 1$, respectively. Then $\lambda = 5t$ and $\lambda = 2t$ are eigenvalues of At.

Here $n = 4$, thus

$$\begin{aligned} r(\lambda) &= \alpha_3\lambda^3 + \alpha^2\lambda^2 + \alpha_1\lambda + \alpha_0 \\ r'(\lambda) &= 3\alpha_3\lambda^2 + 2\alpha_2\lambda + \alpha_1 \\ r''(\lambda) &= 6\alpha_3\lambda + 2\alpha_2 \end{aligned}$$

Since $\lambda = 5t$ is eigenvalue of multiplicity 3 $\Rightarrow e^{5t} = r(5t) = r'(5t)$ and $e^{5t} = r''(5t)$. Thus

$$e^{5t} = \alpha_3(5t)^3 + \alpha_2(5t)^2 + \alpha_1(5t) + \alpha_0 \tag{199}$$

$$e^{5t} = 3\alpha_3(5t)^2 + 2\alpha_2(5t) + \alpha_1 \tag{200}$$

$$e^{5t} = 6\alpha_3(5t) + 2\alpha_2 \tag{201}$$

$$\text{also } \lambda = 2t \quad \Rightarrow e^{2t} = \alpha_3(2t)^3 + \alpha_2(2t)^2 + \alpha_1(2t) = \alpha_0 \tag{202}$$

Thus Equations (199)-(202) are 4 equations in 4 unknowns α_0, α_1, α_2, α_3 therefore $e^{At} = \alpha_3 A^3t^3 + \alpha_2 A^2t^2 + \alpha_1 At + \alpha_0 I$ can be calculated. □

Matrix Polynomials and the Cayley-Hamilton Theorem

Let A be an $n \times n$ matrix with constant entries denote λ_i and u_i be the associated eigenvalues and right eigenvectors, so that

$$Au_i = \lambda_i u_i \qquad i = 1, 2 \ldots n \tag{203}$$

here $\lambda_i \in \mathcal{C}$ is the i^{th} eigenvalue u_i is the i^{th} eigenvector with components $(u_i^1, u_i^2, u_i^3, \cdots u_i^n)^T$.

Premultiply (203) by A

$$A^2 u_i = \lambda_i A u_i = \lambda_i(\lambda_i u_i) = \lambda_i^2 u_i$$

In fact, premultiplying (203) by A^{m-1} shows that A^m has eigenvalues λ_i^m and eigenvectors u_i i.e.

$$A^m u_i = \lambda_i^m u_i$$

Note: One can use a similar argument to show that A^T has the same eigenvalues as those of A:

$$\det(\lambda I = A^T) = \det(\lambda I - A^T)^T = \det(\lambda I - A)$$

$\therefore$ the characteristic equation for A and A^T are the same.

Let $p(\lambda) = \lambda^r + p_1\lambda^{r-1} + p_2\lambda^{r-2} \cdots p_{r-1}\lambda + pr$

the au arbitrary polynomial of degree r. Hence, for the matrix A of size $n \times n$

$$p(A) = A^r + p_1 A^{r-1} \cdots p_r - 1A + prI$$

where I is the identity matrix of size $n \times n$. If u_i are eigenvectors of A then

$$p(A)u_i = A^r + u_1 + p_1 A^{r-1} u_i + \cdots p_r u_i = p(\lambda_i)u$$

showing that the e'values and the e'vectors of $p(A)$ ARE $p(\lambda_i)$ and u_i for $i - 1, 2 \cdots n$.

Cayley-Hamilton Theorem: Every matrix satisfies its own characteristic equation, i.e.

$$k(A) = A^n + k_1 A^{n-1} + k_2 A^{n-2} \cdots k_n I = 0$$

ex) $A = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \Rightarrow$ characteristic polynomial is $\lambda^2 - 3\lambda - 4 = 0$

So $k(A) = A^2 - 3A - 4I \equiv 0$

$$\begin{bmatrix} 7 & 9 \\ 6 & 10 \end{bmatrix} - 3\begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} - 4\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Now, consider e^{tA} (A is constant entry $n \times n$ matrix): it satisfies

$$\frac{d}{dt}e^{tA} = Ae^{tA}$$

$$\text{and } \frac{d^k}{dt^k}e^{tA} = A^k e^{tA} \quad k \geq 0$$

In fact for every polynomial p

$$p\left(\frac{d}{dt}\right)e^{tA} = p(A)e^{tA}$$

the solution of $p\left(\frac{d}{dt}\right)z = 0$ is $z = \sum_{j=1}^{n} c_j z_j(t)$

where $\{c_j\}_1^n$ are constant coefficients. Similarly

$$e^{tA} = \sum_{j=1}^{n} C_j z_j(t) \tag{204}$$

$\{C_j\}_1^n$ are constant matrices, derived by taking derivatives of (204) with respect to t and evaluating them at $t = 0$. The k^{th} derivatives of (204) is

$$A^k = \sum_{j=1}^{n} C_j y_j^{(k)}(0). \tag{205}$$

If the independent solutions $\{y_j\}_{j=}^n$ are chosen to satisfy $y_j^{k-1}(0) = \delta_{jk}$

$\Rightarrow$ from (205) $e^{tA} = \sum_{j=1}^{n} A^{j-1} yj(t)$.

Moreover, if p has simple roots and the $\{yj\}_{j=1}^n$ are chosen to be

$$y_j^{(t)} = e^{\lambda_j t}, \quad \lambda_j \text{ the roots of } p,$$

then (205) becomes the set of n equations

$$A^k = \sum_{j}^{n} = C_j \lambda_j^k \quad k = 0, 1 \ldots n-1$$

which can be solved for $\{C_j\}_{j=1}^n$ which are spectal projections corresponding to the e'values $\{\lambda_j\}_{j=1}^n$.

□