FOX LECTURE NOTES

Ph129: NUMERICAL ANALYSIS

E: Numerical Analysis

References are:

General: Mathews and Walker, chapter 13. (as usual, reasonable introduction to subject)

M. Abramowitz and I. Stegun: "Handbook of Mathematical Functions": this book has already been recommended for theory of special functions; it is also vital when calculating same as it has both formulae and actual values to check your subroutine with.

B. Noble: "Numerical methods" (Oliver and Boyd)

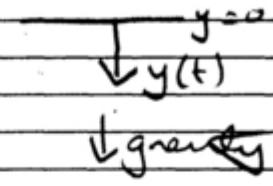
NPL: "Modern computing methods"

are oldies but goodies. Noble is more complete but some of NPL chapters are very well written.

L. Fox (the oxford ffoxe) and D.F. Mayers "Computing methods for scientists and Engineers": Elementary but not very good unless you want lengthy account of philosophy as opposed to facts.

Physics Problem:

p34 Gurlot



m × acceleration

= Force.

Force = mg − drag

drag = $k_n v^n$ $\quad n = 1, 2$

pebble

Force = $mg\left(1 - v^2/v_2^2\right)$ $\quad v_2 = 30$ meters/sec

$g = 9.8$ meters/sec^2

Units: Brower Story.

$$\ddot{y} = g\left(1 - v^2/v_2^2\right) = F(v)$$ (generally $F(y, v)$)

convert to a system of first order ODE's

$$dy/dt = v$$

$$\frac{dv}{dt} = F(y, v)$$

Divide time into equal intervals

$\Delta t \rightarrow$

0 1 2 3 4 5 6 7 8 9

$$t_n = t_0 + n\Delta t$$

any function $g(y, \quad , t) \equiv g(t)$

$$g_n \equiv g(t_n)$$

Euler $\quad \left. \frac{dg}{dt} \right|_{t=t_n} = (g_{n+1} - g_n)/\Delta t$

$$y_{n+1} - y_n = v_n \Delta t$$

$$v_{n+1} - v_n = F(v_n)\Delta t$$

Like statistics, numerical analysis is a despised subject and lot of computational problems are set up incorrectly and much computer / human time wasted. I will try to give a feeling for the methods you should be using when setting up a numerical problem: you can only master them by practical experience.

1. Introduction
2. Matrix Problems - Inversion
   - Eigenvalues
3. Polynomials
4. Calculation of Functions
   - Difference methods: interpolation, extrapolation.
   - Chebyshev Series
   - Regression.
5. Integration
   - 1 dimensional : Gauss, Simpson, Romberg
   - n " " : Monte Carlo
6. Ordinary Differential Equations
7. Partial Differential Equations
   - Elliptic
   - Parabolic
   - Hyperbolic.

E1: Introduction (Things to keep you awake at night).

Suppose we have to solve numerically the recurrence relation:

$$f_{r+1} = 8f_r - 12f_{r-1} \qquad (E1.1)$$

plus the boundary conditions

$$f_1 = 1 \quad f_2 = 2 \qquad (E1.2)$$

Then this is a splendid well posed <u>mathematical</u> problem and the general solution is

$$f_r = 2^r \qquad (E1.3)$$

and it was not necessary to come to Caltech to find this out.

However suppose we had taken

$$f_1 = 1 \quad f_2 = 2.01 \qquad (E1.4)$$

Then solving (E1.1) gives

	$r =$	1	2	3	4	5	6
(E1.4)		1	2.01	4.08	8.52	19.2	51.36
(E1.2)		1	2	4	8	16	32

and very soon, the approximate solution bears no resemblance to the real

one. The reason for this is that - until you impose boundary conditions - general solution of (E1.1) is:

$$f_r = A.2^r + B.6^r$$: the divergence between true and approximate solution is just a reflection of fact that any deviation from (E1.1,2) (e.g. in inexact boundary conditions (E1.4) or from rounding error when solving (E1.1)) will pick up a little $B \neq 0$ and immediately 6^r will propogate through (E1.1) and cause divergence we saw in example. This illustrates two points:

(a) if (E1.1) was an equation of physics and (E1.2) measured experimental data for which we get

$$f_1 = 1 \pm .01 \qquad f_2 = 2.01 \pm .01$$

Then (E1.4) shows that the problem of finding f_{20} is insoluble and the numerical calculation impossible.

(b) If (E1.1, 2) are both laws of physics, then problem ~~is problem~~ is now soluble mathematically but just poorly set up numerically and impossible to solve on any computer which has a finite number of digits and will give rounding errors to propagate as 6^r. In this case one must use mathematical ingenuity to recast problem in a form which is numerically stable and so rounding errors will not cause divergence. for instance $6f_{r+1} = 13 f_r - 2 f_{r-1}$ (E1.5) with same boundary conditions (E1.2) has general solution $A 2^r + B 6^{-r}$ and any $B \neq 0$ will disappear and not increase ad nauseam.

Alternately $f_{r+1} = 2 f_r$, $f_0 = 1$ (E1.6) is an even simpler reformulation which is mathematically identical but numerically distinct from (E1.1).

(ii) (a) Similar difficulties occur with matrices. For instance, suppose we wish to solve:

$$\begin{aligned} x + 2y &= 4 + R_1 \\ x + 2.001y &= 4.003 + R_2 \end{aligned} \qquad (E1.6)$$

Then for $R_1 = R_2 = 0$, we have exact solution $y = 3$ $x = -2$. However the small change in rhs (which might be data again) to $R_1 = 0$ $R_2 = -.006$ changes solution drastically to $x = 10$ $y = -3$. Thus (E1.6) is very unstable to rounding and measurement errors.

For (E1.6) the source of the difficulty is quite obvious - the lhs's are almost identical - alternately we can phrase it as the determinant of matrix $\begin{bmatrix} 1 & 2 \\ 1 & 2.001 \end{bmatrix}$ being 0.001 and much smaller than values of elements (cofactors).

(b) A less blatant example is the notorious Hilbert matrix which is so pathological it is standard test case to prove out matrix inversion routines.

In 4 dimensions:

$$A = \begin{bmatrix} 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \\ 1/5 & 1/6 & 1/7 & 1/8 \end{bmatrix}$$

$$A^{-1} = \begin{bmatrix} 200 & -1200 & 2100 & -1120 \\ -1200 & 8100 & -15120 & 8400 \\ 2100 & -15120 & \underline{29400} & -16800 \\ -1120 & 8400 & -16800 & 9800 \end{bmatrix}$$

Solve $Ax = b$ with $b_1 = b_2 = b_3 = b_4 = 1$.

Then
$$\begin{aligned} x_1 &= -20 \\ x_2 &= 180 \\ x_3 &= -420 \\ x_4 &= 280 \end{aligned}$$

However elements of A^{-1} are very large - maximum is 30,000. So error in x_i

$\delta x \approx 30{,}000\ \delta b$ where δb error in b. This can clearly be very large compared to value of x_i even for modest (10^{-3}) errors in b.

(c) One can formulate these difficulties, rather neatly with eigenvalues ~~of~~ λ of A.

Let $c = |\lambda_{max}| / |\lambda_{min}|$

If $c \approx 1$ problem well behaved.

$c \gg 1$ problem (matrix) ill-conditioned.

Thus $Ax = b$

$A = P^{-1} D_g P \qquad P^{-1} = P^T$

(A symmetric, D_g diagonal) with elements = eigenvalues λ_i)

$b' = Pb \quad , x = P^T y$

$D_g \, y = b'$

and $\delta y_i = \delta b'_i / \lambda$

If λ very small, (compared to λ_{max}) then a small error in b' ($\delta b'_i$) will give a large error in y and problem is poorly posed.

However this illustrates another point. Suppose we do have a poorly posed problem $c \gg 1$, then we needn't

throw up our hands. Although <u>all</u> the x's are undetermined (as $x = P^T y$ is l.c. of all y's and one y at least badly determined) we can state the numerical solution neatly in terms of y's. i.e.

Not all <u>of</u> n physical parameters x_i are determined by our measurements of b_i, rather m ($<n$) l.c.'s of x y_i (with large eigenvalues) are well determined and $n-m$ l.c.'s are badly determined. The latter require a new experiment.

(d) In any numerical calculation, one has to be very careful about undetermined parameters because computer will give answers for all. One has to <u>either</u> check theoretically problem well-posed <u>or</u> check numerically that |final answer| not << |individual terms of sum|.

E2: Matrix Problems

E2/1 Solution of Linear Equations

(i) There is nothing subtle about basic method – Gaussian elimination with interchanges – its what you do when solving by hand

(a) $a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$ (E2.1)

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$ (E2.2)

$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$ (E2.3)

First eliminate x_1 from last two

$-(1) \times a_{21}/a_{11} + (2) \Rightarrow (2')$

$-(1) \times a_{31}/a_{11} + (3) \Rightarrow (3')$

Now eliminate x_2 between (2') and (3')

$-(2') \times a'_{32}/a'_{22} + (3') \Rightarrow (3'')$

and altogether

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a'_{22}x_2 + a'_{23}x_3 = b'_2 \quad \text{(E2.4)}$$
$$a''_{33}x_3 = b'_3$$

and we solve these in obvious way

from bottom up.

We can represent the last equations as $\quad Ux = b'$

where U is upper triangular $\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix}$

while we can represent the steps creating U from A as action of L a lower triangular $\begin{bmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \end{bmatrix}$ matrix.

So method is:

$$\begin{aligned} LA &= U \\ Lb &= b' \\ \Rightarrow Ux &= b' \end{aligned} \qquad (E2.5)$$

(b) Note that this procedure is much more efficient than first calculating inverse A^{-1} <u>even</u> when you have many right hand sides b.

Thus if we wish to solve.

$$Ax = b \qquad \begin{matrix} A & n \times n \\ x, b & n \times m \end{matrix}$$

i.e. n linear equations with m right hand sides

Then Gaussian elimination needs

$$n^3/3 - n/3 + mn^2 \quad \text{operations}$$

while finding A^{-1} and then solving for x by simple multiplication, needs

$$n^3 + mn^2 \quad \text{operations}$$ – this is always more than Gaussian number.

(c) Although the method described in (a) is mathematically complete, it needs numerical refinement to be numerically satisfactory. Thus mathematically if $a_{11} \equiv 0$ then the procedure breaks down: in numerical application, we just need a_{11} to be small and then errors are proportional to $\delta b_1 / a_{11}$ and can be large. It is clearly good to make a_{11} as large as possible. This is done in the "method of maximal column pivots" i.e. at kth step find i_k such that $|a_{i_k k}|$ is maximum for $k+1 \le i_k \le n$. This is maximum coefficient

of x_k in equations $k+1 \ldots n$ and this use ensures that $a_{lk}/|a_{kk}|$ (the multipliers when eliminating x_k) are all ≤ 1 in modulus and hence minimizes effect of rounding error.

(d) Rounding error

We can estimate the effect of rounding error by the important technique of backward error analysis. In this, rather than directly estimating error in computed solution $\bar{x}$, we try to find equation that it satisfies

$$(A+\delta A)\bar{x} = b+\delta b$$
$$A\,x = b \qquad \text{(E2.6)}$$

Here x is exact and $\bar{x}$ computed solution. Of course $\delta A, \delta b$ are not unique - however the analysis will find particular values of them.

First assume that rows of A have been permuted to ensure that maximal pivot has $L_k = k$. Then at k'th step, we must eliminate x_k from equations $k+1 .. n$ and before that, write equations

$$A^{(k)} x^{(k)} = b^{(k)}$$

where if there were no rounding error, $x^{(k)}$ is independent of k. $A^{(1)}$, $b^{(1)}$ are original matrix and r.h.s : $x^{(1)} = x$ is true solution. At k'th stage, we choose $a_{kk}^{(k)}$ as a pivot to generate $A^{(k+1)}$ which has identical first k rows to $A^{(k)}$.

So if we put $y = x^{(n)}$, it satisfies:

$$\begin{aligned} a_{11}^{(1)} y_1 + a_{12}^{(1)} y_2 + a_{13}^{(1)} y_3 + a_{14}^{(1)} y_4 &= b_1^{(1)} \\ a_{22}^{(2)} y_2 + a_{23}^{(2)} y_3 + a_{24}^{(2)} y_4 &= b_2^{(2)} \\ a_{33}^{(3)} y_3 + a_{34}^{(3)} y_4 &= b_3^{(3)} \\ a_{44}^{(4)} y_4 &= b_4^{(4)} \end{aligned} \quad (E2.7)$$

Let g_k be maximum $a^{(l)}$ that ever appears.
" " $b^{(l)}$ " " "
and η be a typical rounding error

Now we want to find difference, between $x^{(n-1)}$ (call it z) and $x^{(n)}$ $(=y)$

z satisfies same first two equations as y plus:

$$a_{33}^{(3)} z_3 + a_{34}^{(3)} z_4 = b_3^{(3)} \qquad (E2.8)$$

$$a_{43}^{(3)} z_3 + a_{44}^{(3)} z_4 = b_4^{(3)} \qquad (E2.9)$$

1st form multiplier:

$$m_{43} = -a_{43}^{(3)} / a_{33}^{(3)} + \text{Round} \qquad (E2.10)$$

$|\text{Round}| \leq \eta\, |m_{43}| < \eta$ as $|m_{43}| \leq 1$.

Then take (E2.9) $+ m_{43}$ (E2.8)

$$\Rightarrow \quad a_{44}^{(4)} = a_{44}^{(3)} + m_{43} a_{34}^{(3)} + \eta g$$

$$b_4^{(4)} = b_4^{(3)} + m_{43} b_3^{(3)} + \eta h$$

and as by definition $a_{44}^{(4)} y_4 = b_4^{(4)}$

$$[a_{44}^{(3)} + \eta g] y_4 + m_{43} a_{34}^{(3)} y_4 = b_4^{(3)} + \eta h + m_{43} b_3^{(3)}$$

while we still have: $a_{33}^{(3)} y_3 + a_{34}^{(3)} y_4 = b_3^{(3)}$

These last two equations are exact & we can multiply last by m_{43} and subtract

$$-m_{43} a_{33}^{(3)} y_3 + [a_{44}^{(3)} + \eta g] y_4 = b_4^{(3)} + \eta h$$

But from (E2.10)

$$-m_{43}\, a_{33}^{(3)} = a_{43}^{(3)} + \eta g$$

and so we can write last equation for y:

$$\left[a_{43}^{(3)} + \eta g\right] y_3 + \left[a_{44}^{(3)} + \eta g\right] y_4 = b_4^{(3)} + \eta h$$

<u>So</u> $$A^{(4)} = A^{(3)} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \eta g$$

$$b^{(4)} = b^{(3)} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \eta h$$

One can continue this process and eventually relate $A^{(4)}$ to $A^{(2)} = A$ †

$$\begin{aligned} A^{(4)} &= A^{(2)} + \delta\tilde{A} \\ b^{(4)} &= b^{(2)} + \delta\tilde{b} \end{aligned} \qquad (E2.20)$$

$$\delta\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \end{bmatrix} \eta g \qquad \delta\tilde{b} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} \eta h$$

with obvious generalization to non matrices.

† A bit tricky: this is <u>an</u> $A^{(4)}, b^{(4)}$ such that $A^{(4)}y = b^{(4)}$: it is not <u>the</u> $A^{(4)}, b^{(4)}$ given by computer – see subtlety bottom of page 18.

Now we not quite finished for y is not final calculated solution as we must still do "back substitution" on equation $A^{(n)} y = b^{(n)}$.

To see what this does - consider a typical calculation - putting $\bar{x}$ as final answer

$$\bar{x}_2 = \frac{b_2^{(2)} - a_{23}^{(2)} \bar{x}_3 - a_{24}^{(2)} \bar{x}_4}{a_{22}^{(2)}} + \frac{\eta h}{|a_{22}^{(2)}|}$$

<u>or</u> $\bar{x}_2$ satisfies exact equation

$$a_{22}^{(2)} \bar{x}_2 + a_{23}^{(2)} \bar{x}_3 + a_{24}^{(2)} \bar{x}_4 = b_2^{(2)} + \eta h$$

or $$A^{(n)} \bar{x} = b^{(n)} + \delta c$$

$$\delta c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \eta h$$

If you redo previous analysis with $b^{(n)} + \delta c$ replacing $b^{(n)}$ (see bottom of page 16 and $|m_{43}| < 1$), you find that it is <u>not</u> correct to add δc to $\delta \tilde{b}$: rather $\bar{x}$ satisfies

$$[A+\delta A]\,\bar{x} = b + \delta b \qquad (E2.6)$$

where $\delta A = \delta\hat{A}$ is as before but

$$\delta b = \begin{bmatrix} 1 \\ 3 \\ 5 \\ \vdots \\ 2n-1 \end{bmatrix} \eta h$$

Now we're home: multiply (E2.6) by A^{-1}

$$\bar{x} = x - A^{-1}\delta A\,\bar{x} + A^{-1}\delta b \qquad (E2.11)$$

which relates calculated to true solution. ~~difference~~ The difference is large if rounding error ηg, ηh large or if A^{-1} has large elements. (i.e. if problem ill-conditioned.

Consider last term in (E2.11): if $\bar{a}$ typical size of A^{-1}, then it contributes ~~max~~ an error to $\bar{x}$ of maximum value:

$$|\bar{x} - x| \approx \bar{a}\, n^2 \eta h$$

while - given statistical rounding error we get $|\bar{x} - x| \approx \bar{a}\sqrt{n^2}\,\eta h$ which doesn't increase that fast with n.

(ii) Triangular Decomposition

In (E2.5) we summarized Gaussian elimination as:

$$LA = U$$
$$Lb = b'$$
$$Ux = b'$$

if L is any lower triangular matrix, its inverse is also lower triangular

$$LL' = I:$$

hence $A = L'U$ expresses A as a product of triangular matrices.

The decomposition

$$\begin{aligned} A &= L'U \\ L'L &= I \\ b' &= Lb \\ Ux &= b' \end{aligned} \qquad (E2.12)$$

is called the method of triangular decomposition: Gaussian elimination is one way of finding such a form (E2.12). It turns out that one can arrange the calculation of L' and U in (E2.12) more economically than in Gaussian elimination and get smaller rounding error. This is

So even if $U \equiv U_{gauss}$. This is described in Fox and Meyers.

(iii) Iterative Methods

(a) Suppose we have some guess $\underline{x}^{(0)}$ for a solution of $A\underline{x} = \underline{b}$

we can obtain a possibly better solution by ~~for~~ solving

$$\begin{aligned} \underline{a_{11}}\ \underline{x_1^{(1)}} + a_{12}\, x_2^{(0)} + \cdots + a_{1n} x_n^{(0)} &= d_1 \\ a_{21}\, x_1^{(0)} + \underline{a_{22}\, x_2^{(1)}} + \cdots + a_{2n} x_n^{(0)} &= d_2 \\ \cdots\cdots\cdots \end{aligned} \qquad (E2.13)$$

This is called Jacobi Iteration.

Alternately we can use our new solution whenever possible.

as before:

$$\begin{aligned} \underline{a_{11}}\ \underline{x_1^{(1)}} + a_{12}\, x_2^{(0)} + \cdots + a_{1n} x_n^{(0)} &= d_1 \\ \underline{a_{21}}\ \underline{x_1^{(1)}} + \underline{a_{22}}\ \underline{x_2^{(1)}} + \cdots\ a_{2n} x_n^{(0)} &= d_2 \\ \cdots \end{aligned} \qquad (E2.14)$$

This is called Gauss-Seidel Iteration

(b) Examine this in general: first permute rows to ensure that maximum coefficients were in diagonal positions

Then divide through to make diagonal elements 1. Now we decompose

$$A = I - L - U \qquad (E2.16)$$

where L, U are lower, upper triangular with zero diagonal elements.

We can state our methods

<u>Jacobi</u>: $\underline{x}^{(n+1)} = (L+U)\underline{x}^{(n)} + \underline{d}$ (E2.17)

<u>Gauss-Seidel</u>: $\underline{x}^{(n+1)} = L\underline{x}^{(n+1)} + U\underline{x}^{(n)} + \underline{d}$

(c) Convergence

Let x be true solution

$$x = (L+U)x + d$$

and $e^{(n)} = x^{(n)} - x$: error at n'th step

<u>Jacobi</u> $\underline{x}^{(n+1)} = (L+U)\underline{x}^{(n)} - (L+U)\underline{x} + \underline{x}$

<u>or</u> $\underline{e}^{(n+1)} = (L+U)\,\underline{e}^{n}$

<u>Gauss-Seidel</u>: $\underline{x}^{(n+1)} = L x^{(n+1)} + U\underline{x}^{(n)} - (L+U)\underline{x} + \underline{x}$

or $(I - L)\,\underline{e}^{(n+1)} = U\,\underline{e}^{(n)}$

in either case

$$\underline{e}^{(n+1)} = C\,e^{(n)}$$

$C = (L+U)$ (Jacobi) or $(I-L)^{-1}U$ (Gauss)

So $\quad \underline{e}^{(n)} = C^{(n)} \underline{e}^{(0)}$

Put $\quad C = P^{-1} \, dg(\lambda_i) P$

$$C^n = P^{-1} \, dg(\lambda_i^n) P$$

and $C^n \to 0$ (i.e. process converges) iff <u>all</u> eigenvalues of C have $|\lambda| < 1$.

This is certainly <u>not</u> always the case and conditions for Jacobi may be different from Gauss-Seidel.

For instance, Gauss-Seidel <u>always</u> converges if A is symmetric and positive definite. Jacobi will only converge if $|\lambda - 1| < 1$ for all eigenvalues λ of A.

(d) Over-Relaxation

We write the general iteration scheme:

$$N_0 x^{(n+1)} = P_0 x^{(n)} + d$$

where $A = N_0 - P_0$ ensures that if it converges $x = x^{(n+1)} \to x^{(n)}$, then solution satisfies $Ax = d$, we get same solution, if

we replace $N_0 \to N_0(\alpha) = N_0(1+\alpha)$
$P_0 \to P_0(\alpha) = P_0 + \alpha N_0$.
as $N_0(\alpha) - P_0(\alpha) = N_0 - P_0$.
The new iteration scheme reads:

$$(1+\alpha) N_0 x^{(n+1)} = P_0 x^{(n)} + \alpha N_0 x^{(n)} + d$$

or $$x^{(n+1)} = \frac{1}{1+\alpha}\left\{N_0^{-1}(P_0 x^{(n)} + d)\right\} + \frac{\alpha x^{(n)}}{\alpha+1}$$

$$= \omega_1 x^{(n+1)}_{old} + \omega_2 x^{(n)}$$

with $\omega_1 + \omega_2 = 1$ i.e. linear combination of old iteration $x^{(n+1)}$ and $x^{(n)}$ with sum of weights $=1$. Old iteration matrix $C = N_0^{-1} P_0$ with eigenvalues λ_i: new iteration matrix is $\frac{1}{\alpha+1}\left[N_0^{-1}P_0 + \alpha\right]$

and has eigenvalues $\mu_i = \frac{\alpha+\lambda_i}{\alpha+1}$.

The idea, of course, is to determine α so that $|\mu_i|_{max} < |\lambda_i|_{max}$ and scheme converges faster. In the problem set, this is done for matrices which satisfy "property A" (see NPL handbook p. 39). Again Isaacson and Keller (page 73 onwards) consider case where $N_0^{-1} P_0$ has eigenvalues

$$\lambda_1 \le \lambda_2 \le \dots < 1$$

where λ_1 may be arbitrarily negative: scheme diverges if $\lambda_1 < -1$. However by choosing $\alpha > -(1+\lambda_1)/2$, new series converges while the best choice ($|p_{max}|$ smallest) is $\alpha_{BEST} = -\frac{1}{2}(\lambda_1 + \lambda_n)$.

There are many variants of relaxation methods - for instance 'property A' in NPL book, is not really of standard form.

Why use iterative methods? Clearly for general matrix, direct Gaussian elimination is preferable as it is essentially guaranteed to converge. However, there are certain classes of matrices - of large order and with many zero elements - for which the iterative methods can be proved to converge: then they are used. This occurs in solution of partial differential equations.

E82/2 Methods for Finding Eigenvalues/vectors of Matrices

(i) Iteration Methods

Let A be any $n \times n$ matrix with eigenvalues λ_i and eigenvectors $\underline{x}_i$. Order them $|\lambda_1| > |\lambda_2| > \ldots$

Define iteration: $\underline{x}^{(k)} = A\,\underline{x}^{(k-1)}$

Initially $\underline{x}^{(0)} =$ Anything.

Expand $\underline{x}^{(0)} = \sum_{i=1}^{n} k_i\,\underline{x}_i$

Then $\underline{x}^{(k)} = \sum_{i=1}^{n} k_i\,\lambda_i^k\,\underline{x}_i$

as $k \to \infty$ $\underline{x}^{(k)} \to k_1\,\lambda_1^k\,\underline{x}_1$

This method automatically gives $\underline{x}^{(k\to\infty)}$ a ~~root of 1~~ eigenvector of largest size. We normalize it and form

$$A_1 = A - \lambda_1\,x_1\,x_1^T$$

which for symmetric matrix A has same eigenvalues/vectors as A except that λ_1 is replaced by 0. For unsymmetric matrices, a similar use works ~~with~~ and it is described p. 26-27 of NPL.

(ii) Actually for symmetric matrices, other methods are always used. These depend on fact that if T is orthogonal, A and TAT^T have same eigenvalues:

i.e. $\det[TAT^T - \lambda I] = \det T[A-\lambda I]T^T$

$= (\det T)^2 \det(A - \lambda I)$

i.e. zeros of determinant unaltered and hence eigenvalues same.

(iii) Jacobi's method

This is the most direct of the methods which try to find a matrix T such that TAT^T is easy to find roots of. It is not used in practice as methods (Givens, Householder) that reduce A to tridiagonal form are faster.

Take $n=4$ and let

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & -\sin\theta \\ 0 & 0 & 1 & 0 \\ 0 & \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$T^T A T = \begin{bmatrix} a_{11} & a_{12}\cos\theta + a_{14}\sin\theta & a_{13} & -a_{12}\sin\theta + a_{14}\cos\theta \\ \cdot & a_{22}\cos^2\theta + 2a_{24}\sin\theta\cos\theta + a_{44}\sin^2\theta & a_{23}\cos\theta + a_{34}\sin\theta & \boxed{a_{24}(\cos^2\theta - \sin^2\theta) + (a_{44}-a_{22})\sin\theta\cos\theta} \\ \cdot & \cdot & a_{33} & -a_{23}\sin\theta + a_{34}\cos\theta \\ \cdot & \cdot & \cdot & a_{22}\sin^2\theta - 2a_{24}\sin\theta\cos\theta + a_{44}\cos^2\theta \end{bmatrix}$$

complete by symmetry

<u>Now</u> $\mathrm{Tr}(T^T A T) = \mathrm{Tr}(A T T^T) = \mathrm{Tr}(A)$ is unchanged.

<u>Choose</u> θ so that new value of a_{24} is zero i.e.

$$a_{24}(\cos^2\theta - \sin^2\theta) + (a_{44} - a_{22})\sin\theta\cos\theta = 0$$

i.e. $$\tfrac{1}{2}\tan 2\theta = a_{24}/(a_{22} - a_{44}) \qquad (E2.18)$$

meanwhile sum of squares of off-diagonal terms

<u>before</u> : $a_{12}^2 + a_{13}^2 + a_{14}^2 + a_{23}^2 + a_{24}^2 + a_{34}^2$

<u>after</u> : $a_{12}^2 + a_{13}^2 + a_{14}^2 + a_{23}^2 + a_{34}^2$

and so it <u>has</u> reduced. So the idea of Jacobi method is to succesively

choose T as a rotation in i,j'th plane where $|a_{ij}|$ is largest off diagonal element. Then sum of squares of off diagonal elements will decrease by $|a_{ij}|^2$ and $\to 0$. So we get:

$$T_k^T T_{k-1}^T \cdots T_1^T A T_1 T_2 \cdots T_k = dg(\lambda_i)$$

and eigenvalues are just columns of $T = T_1 T_2 \cdots T_k$.

Note Jacobi's method, although it cannot diverge, does not have a definite number of steps after which you have answer.

(iv) Givens

Better are the methods of Givens and Householder which first reduce matrix to tridiagonal $\begin{bmatrix} x & x & 0 & 0 & 0 \\ x & x & x & 0 & 0 \\ 0 & x & x & x & 0 \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix}$ form and then find eigenvalues of this by special methods.

In Givens method, we take same form for T as in Jacobi, but rather

than (E2.18), pick θ so that $a_{14}=0$. This requires $a_{14}/a_{12} = \tan\theta$ and it is crucial to note that if $a_{13}=0$, this fact is preserved. So Givens method is to set

$$a_{(i,j)} = 0 \quad \text{by rotation in} \quad \text{plane} =$$

$$(i,j) = \begin{array}{cccccc} (1,3) & (1,4) & \cdots & (1,n) & (2,3) \cdots & (2,n) \\ & (2,4) & \cdots & (2,n) & (3,4) & (3,n) \end{array}$$

$$(n-2,n) \qquad (n-1,n) \quad \text{etc.}$$

This is done successively and zeroes are preserved: one gets to bidiagonal form faster than Jacobi gets to diagonal form. Note in particular, that number of rotations $\sum_{i=1}^{n-2} (n-i-1) = \frac{1}{2}(n-1)(n-2)$ is finite.

(v) Householder's Method

This is a factor of 2 faster than Given's method and achieves this by a different choice of rotations T.

$$T_{ij} = 1 - 2\,\omega_i \omega_j \quad \text{with } \sum_{i=1}^{n} \omega_i^2 = 1$$

$$\text{or } \underline{\underline{T}} = 1 - 2\,\omega\,\omega^T \qquad \omega^T\omega = 1.$$

T is symmetric and also orthogonal for:

$$T^2 = TT^T = 1 - 4\omega\omega^T + 4\omega\underbrace{\omega^T\omega}_{1}\omega^T = 1.$$

We need only $n-2$ stages and at the k'th stage we set all elements of A which are in k'th row/column, but not in tridiagonal positions, to zero. We choose ω to have zeroes in 1st $\underline{\underline{k}}$ positions and clearly this preserves zeroes created by previous ω transformations of earlier stages.

It is sufficient to consider $k=1$ when

$$\omega = (0, \omega_2, \omega_3 \ldots \omega_n).$$

$$A \rightarrow TAT$$

here $TAT = A - 2w(w^T A) - 2(Aw)w^T$
$+ 4\, w w^T (w^T A w)$

Both the last and the second term automatically give zeroes in the first row so we need only consider first and third terms, If we let b be the vector Aw, then l'th element of new first row is:

$\tilde{a}_{1l} = a_{1l} - 2 b_1 w_l$

and $\tilde{a}_{1l} = 0$ for $l \geq 3$ if:

$$w_l = a_{1l}/2b_1 \qquad l \geq 3 \qquad (E2.19)$$

and w_2 determined by $w^T w = 1$. (E2.20)

(E2.19), (E2.20) are splendid except that b_1 is linear in w and so they are quadratic equations. Actually some algebra shows everything to be very simple. Thus

$$b_1 = w_2 a_{21} + \dots + w_n a_{n1} \qquad (E2.21)$$

Put $$x = \sum_{l \geq 3} a^2_{l1} \qquad (E2.22)$$

$$x' = \sum_{l \geq 2} a^2_{l1} \qquad (E2.23)$$

In (E2.22) replace a_{l1} by $2b_1 w_l$

$$x = 4b_1^2 \sum_{l=3}^{n} w_l^2 = 4b_1^2(1-w_2^2) \qquad (E2.24)$$

Again in (E2.21) replace w_l, $l>2$ by (E2.19).

$$x = b_1 = w_2 a_{21} + \sum_{l\geq 3} a_{1l}^2/2b_1$$

or $\quad 2b_1^2 - 2b_1 w_2 a_{21} = x$

and subtracting $\frac{1}{2}$(E2.24) gives

$$2b_1^2 w_2^2 - 2b_1 w_2 a_{21} = x/2$$

Complete Square:

$$[2b_1 w_2 - a_{21}]^2 = x'$$

or $\quad 2b_1 w_2 - a_{21} = \pm x'$

remember $\quad 2b_1 w_l - a_{l1} = 0 \quad , l\geq 3$ $\quad\Big\}$ (E2.25)

Take (E2.25) and multiply by w_l, and sum from $l=2$ to n - use (E2.20), (E2.21)

$$2b_1 - b_1 = \pm w_2 x'$$

or $\quad b_1 = \pm x' w_2 \qquad (E2.26)$

Substitute (E2.26) in $l=2$ part of (E2.25)

$$w_2^2 = \tfrac{1}{2}\left\{1 \pm a_{12}/x'\right\} \qquad (E2.27)$$

(E2.26,27) now determine b_1 and w_2, whence (E2.19) gives other elements of w.

The $\pm$ sign in (E2.27) is chosen to make ω_2 as big as possible. ($\pm a_{12} > 0$).

(vi) Note that these methods - if applied to unsymmetric matrices, reduce them to Hessenberg form

$$\begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$

For unsymmetric matrices, one still only uses iterative method if one (a few) eigenvalues wanted. The other methods are not known to me.

(vii) Eigenvalues vectors of Tridiagonal Matrix

The best ways of treating such matrices - the proud result of Givens or Householder - are obscure algorithms called QR and LR method. We describe simple method which is pretty. Consider the equation $\det(A - \lambda I) = 0$ satisfied by eigenvalues.

$$\det \begin{bmatrix} \alpha_1 - \lambda & \beta_2 & 0 & 0 \\ \beta_2 & \alpha_2 - \lambda & \beta_3 & 0 \\ 0 & \beta_3 & \alpha_3 - \lambda & \beta_4 \\ \vdots & & & \end{bmatrix} = 0.$$

Define the leading principal minors as

$p_0(\lambda) = 1$

$p_1(\lambda) = \alpha_1 - \lambda$

$p_r(\lambda) = (\alpha_r - \lambda)\, p_{r-1}(\lambda) - \beta_r^2\, p_{r-2}(\lambda)$ (E2.28)

The p_r's have same property as orthogonal polynomials i.e.

<u>Theorem</u>: The p_r form a Sturm sequence and zeroes interlace.

<u>Proof</u>: use induction and assume it is true for $r = i-1$ and $i-2$. Let λ_a, λ_b be two adjacent zeros of $p_{i-1}(\lambda)$. Then by inductive assumption $p_{i-2}(\lambda_a) = -p_{i-2}(\lambda_b)$

Note that p_{i-1} and p_{i-2} cannot have

common zero because otherwise recurrence relation (E2.28) would imply $p_{i-3} \cdot p_{i-4} \cdots p_0$ would have zero which is clearly ridiculous. Then (E2.28) $\Rightarrow$

$$p_i(\lambda_a) = -\beta_i^2 \, p_{i-2}(\lambda_a)$$

$$p_i(\lambda_b) = -\beta_i^2 \, p_{i-2}(\lambda_b)$$

whence $p_i(\lambda_a)$ and $p_i(\lambda_b)$ have opposite signs and so p_i has one (3, 5 ... ?) zeroes between those of p_{i-1}. However examine end zero of p_{i-1}. Then p_i and p_{i-2} have opposite sign there: but for large $\pm\lambda$, ~~they are~~ $p_r(\lambda)$ is proportional to $(-\lambda)^r$ and hence p_i and p_{i-2} have same sign. Thus p_i has one more zero outside those of p_{i-1} and zeroes must exactly interlace.

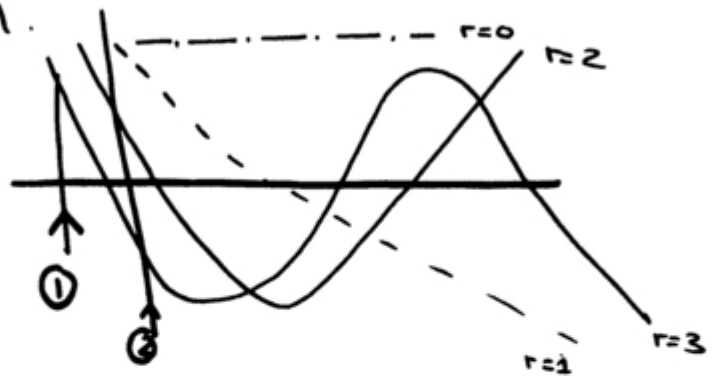
Q.E.D.

Now there is a simple technique to find the eigenvalues. First note that if I take any λ and examine sequence $p_0(\lambda)\ p_1(\lambda) \ldots p_n(\lambda)$, then the number

of eigenvalues with value $< \lambda$ is equal to number of sign changes in sequence

$p_n(\lambda) : p_{n-1}(\lambda) \cdots 1$

<u>Proof</u>: Eigenvalues are just zeroes of $p_n(\lambda)$.



At $\lambda = -\infty$, they all same sign (positive).

Now if I cross zero of $p_n(\lambda)$, I change from p_n and p_{n-1} having same sign to p_n and p_{n-1} having opposite signs. There is net increase in 1 in number of sign changes in sequence.

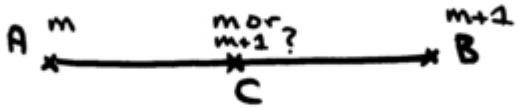
If I cross zero of p_r, $r < n$, then

before p_{r+1} opp. sign to p_r
p_{r-1} same " to p_r

after p_{r+1} same sign as p_r
p_{r-1} opp. sign to p_r

The number of sign changes in sequence does not change Q.E.D.

Now we can just use method of bisections to find eigenvalues. Suppose we know there are m sign changes at A and m+1 at B



Consider midpoint C of AB. It must have either m or m+1 sign changes. If m, zero is in CB; if m+1 in AC. This way I continually bisect and hunt down all eigenvalues.

Note $p_r(\lambda)$ are found quickly from recurrence relation (E2.28) for each λ.

The eigenvectors can be found quite easily - given value λ - this is described in NPL page 32.

E3 Zeros of Functions

E3/1 Polynomials

Mathews and Walker describe Horner's method of finding zeroes of polynomial

$$f(x) \equiv a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 = 0 \qquad (E3.1)$$

However I won't discuss this as it both not used very much and is also rather sensitive to small changes in a_n (in particular to rounding error). To consider effect of small change in a given coefficient a_s on a zero at $x = \alpha$, we differentiate:

$$f(\alpha) = 0 \qquad \text{or} \qquad f(x) = f'(\alpha)(x-\alpha) + \dots$$

$$\cancel{\Rightarrow f'(\alpha)\, \partial\alpha/\partial a_s = x^s}$$

$$\partial f(x)/\partial a_s \Big|_{x=\alpha} = \alpha^s \qquad \text{from (E3.1)}$$

$$= -\,\partial\alpha/\partial a_s \; f'(\alpha) \text{ from above}$$

So $\partial\alpha/\partial a_s = -\,\alpha^s/f'(\alpha)$

If $f'(\alpha)$ small, α is very sensitive to a_s : in particular at a double root $f'(\alpha)=0$, $\partial\alpha/\partial a_s = \infty$ As a contrived example

put: $f(x) = (x+1)(x+2)\ldots(x+20)$ (E3.2)

$\alpha = -16$, $s = 19$

Then $\delta\alpha/\alpha = 3.2 \times 10^{10}\ \delta a_s/a_s$

i.e. we must work with 10 more digits in intermediate steps than we want in final answer <u>whatever</u> method employed to solve for roots of polynomial

Note that if you want to find eigenvalues of a matrix A

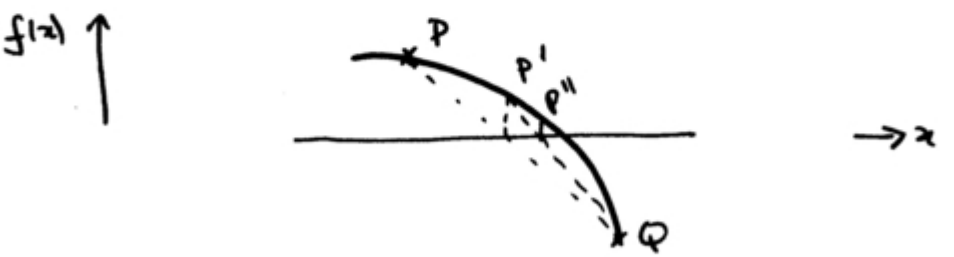
i.e. $\det(A-\lambda I)=0$

which if A is $n \times n$, can be written out as a polynomial (E3.1) in λ. However the latter is much more sensitive to rounding error than original problem. So one should not do this but rather use methods described in §§E2.

E3/2 $f(x) = 0$

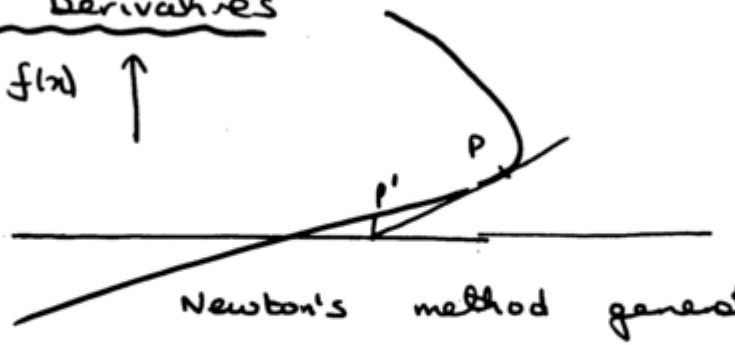
To solve the general problem $f(x)=0$ is one variable special case of problem discussed in statistics of minimizing a general function $g(x_1, x_2 \ldots, x_n)$ of n variables $[f = g']$. The single variable case is in fact quite easy and almost any sensible iterative method with suitable checks to ensure convergence is OK. This is discussed in Mathews and Walker pages 359 → 360.

(i) No Derivatives



The simple bisection method which generates sequence $P', P'' \ldots$, once given two points P and Q straddling zero works.

(ii) Derivatives



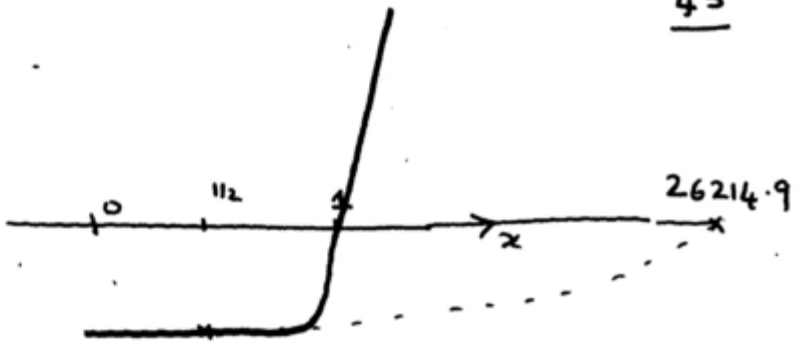
Newton's method generates P' from P using tangent. Mathematically.

$$x(P) \rightarrow x + dx(P')$$

$$dx = -f(x)/f'(x)$$

The derivative method takes fewer (call it d) steps than method (i) (call it i). but $d < i$ $i < 2d$ - So if derivative takes ≈ same time as function to calculate, then one should use non derivative methods (Exactly, for Newton, error → O(error²) each iteration while for interpolation (i), error → O(error^1.62)

- See NPL book page 56, 57).



Suppose we wanted to find root $x=1$ of $f(x) \equiv x^{20} - 1$

Initial Try $x = 1/2$

Newton $\Rightarrow$ next guess as $x = 26214{\cdot}9$.

If we continued to iteration process would in fact converge as

$$x_{r+1} = x_r + (1 - x_r^{20})/20x_r^{19}$$

$$\approx 19x_r/20 \quad \text{for large } x_r$$

This sort of disaster can be avoided by forcing either step $x_{r+1} - x_r$ or function change be < than some suitable amount.

E4 Calculation and Representation of Functions

E4/1 Difference Operators

(i) Difference operators are not as important now as in good old days because on digital computers, functions are evaluated by summing series and not by table look-up (the latter takes too much ~~space~~ core). However they are still important for linear and partial differential equations.

Mathews and Walker introduce them on page 345. Construct a table of $\sin x$ as on following page. $\sin x$ is a typical (analytic) function whose differences get smaller until they become random ~~things of same~~ size as rounding error - the latter then diverge (we give an example of this later under propagation of errors - it is not shown in $\sin x$ table).

	x	$y = \sin x$				
y_{-4}	0°	0				
			·1736			
y_{-3}	10°	·1736		−·0052		
			·1684		−·0052	
y_{-2}	20°	·342		−·0104		·0004
			·1580		−·0048	0
y_{-1}	30°	·5		−·0152		·0004
			·1428		−·0044	·0004
y_0	40°	·6428		−·0196		·0008
			·1232		−·0036	−·0003
y_1	50°	·766		−·0232		·0005
			·1		−·0031	·0003
y_2	60°	·866		−·0263		·0008
			·0737		−·0023	·0002
y_3	70°	·9397		−·0286		·0010
			·0451		−·0013	
y_4	80°	·9848		−·0299		
			·0152			
y_5	90°	1.0				

↑ 1st 2nd differences of values in previous column.

↑ rounding errors

(ii) The numbers in this table can be labelled according to 3 different schemes. Let $y_n = (x_0 + nh)$ where h is some fixed interval and n an integer.

(a) Forward Differences

$$\Delta y_n = y_{n+1} - y_n \quad \text{def.}$$

e.g. $\Delta y_0 = y_1 - y_0 = \cdot 1232$ above

$$\Delta^2 y_n = \Delta y_{n+1} - \Delta y_n \quad \text{def.}$$

e.g. $\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = -.0232$ above

The pattern is

$$
\begin{array}{ccccc}
y_{-2} & & & & \\
 & \Delta y_{-2} & & & \\
y_{-1} & & \Delta^2 y_{-2} & & \\
 & \Delta y_{-1} & & \Delta^3 y_{-2} & \\
y_0 & & \Delta^2 y_{-1} & & \Delta^4 y_{-2} \\
 & \Delta y_0 & & \Delta^3 y_{-1} & \\
y_1 & & \Delta^2 y_0 & & \\
 & \Delta y_1 & & & \\
y_2 & & & &
\end{array}
$$

← Same subscript on diagonals

(b) Backward Differences

$$\nabla y_{n+1} = y_{n+1} - y_n \quad \text{def.}$$

$\nabla y_{n+1} = \Delta y_n$ of course, and pattern is

$$
\begin{array}{ccccc}
y_{-2} & & & & \\
 & \nabla y_{-1} & & & \\
y_{-1} & & \nabla^2 y_0 & & \\
 & \nabla y_0 & & \nabla^3 y_1 & \\
y_0 & & \nabla^2 y_1 & & \nabla^4 y_2 \\
 & \nabla y_1 & & \nabla^3 y_2 & \\
y_1 & & \nabla^2 y_2 & & \\
 & \nabla y_2 & & & \\
y_2 & & & &
\end{array}
$$

← Same subscript on diagonal.

(c) Central Differences

$$\delta y_{n+1/2} = y_{n+1} - y_n$$

y_{-2}				
	$\delta y_{-3/2}$			
y_{-1}		$\delta^2 y_{-1}$		
	$\delta y_{-1/2}$		$\delta^3 y_{-1/2}$	
y_0		$\delta^2 y_0$		$\delta^4 y_0$
	$\delta y_{1/2}$		$\delta^3 y_{+1/2}$	
y_1		$\delta^2 y_1$		
	$\delta y_{3/2}$			
y_2				

← Same numbers on horiz lines but only appear every other order.

So δy_0 is not on table but only $\delta^{2n} y_0$. (n integer)

(iii) Two Small points

(a) Polynomials

Note that like the $(m+1)$'th derivative, the $(m+1)$'th. differences of a polynomial of degree m are zero. One can prove this by expressing $y_n = f(x_0 + nh)$ as power series in n. Then it is easy to see that one goes down one in highest power of h each difference. Alternatively

the formal methods we will soon discuss, prove this immediately.

(b) Error Propagation

Suppose we add to $f(x)$ an error in one entry. Then superimposed on differences of $f(x)$, we will have pattern.

Error					
0					
0					
0				1	
			1		
0		1		-4	a fan-out :
	1		-3		
1		-2		6	this can be used
	-1		3		
0		1		-4	to detect errors.
			1		
0				1	Note rounding
		Binomial Coefficients			
0					errors will give this
0					as they are a special
					sort of error !

(iv) Symbolic Methods

(a) Define $E y_n = y_{n+1}$

$$\mu y_n = \tfrac{1}{2}\left(y_{n+\frac{1}{2}} + y_{n-\frac{1}{2}} \right)$$

where E = displacement $\quad$ " $y(x_0 + (n+\tfrac{1}{2})h)$ see below.
μ = averaging operator

put $\quad E^s y(x) = y(x+sh)$ $\quad$ any s, not necessarily integer.
$\uparrow$ fixed.

Note this is consistent as $E^{s_1} E^{s_2} = E^{s_1+s_2}$

Then $\mu = \tfrac{1}{2}\left(E^{\frac{1}{2}} + E^{-\frac{1}{2}} \right)$.

we can now express difference operators

$$\Delta = E - 1$$
$$\nabla = 1 - E^{-1}$$
$$\delta = E^{\frac{1}{2}} - E^{-\frac{1}{2}}$$

(b) more interestingly, we can relate the differential operator D to E and then to Δ, ∇ and δ. This will be important for differential equations.

$$Dy = dy/dx \qquad \text{def.}$$

$$E y(x) = y(x+h) \qquad \text{- use Taylor}$$
$$= y(x) + h y' + h^2/2!\, y'' + \dots$$
$$= \left(1 + hD + h^2 D^2/2! + \dots\right) y(x)$$

or $E\, y(x) = \exp(hD)\; y(x)$

So $E = \exp(hD)$

Taking $\delta = E^{1/2} - E^{-1/2}$

we get $\delta = \exp(\tfrac{1}{2}hD) - \exp(-\tfrac{1}{2}hD)$

or $\delta = 2 \sinh \tfrac{1}{2}hD$

or $hD = 2 \sinh^{-1} \tfrac{1}{2}\delta$

Taylor expanding $hD = \delta - \delta^3/24$ (using $\sinh x = x - x^3/6$)

Similarly $hD = \log(1+\Delta)$

$= \Delta + \Delta^2/2$ but this converges more slowly than central difference formula.

Note that $\delta^{(m)} \approx h^m D^m$

i.e. $\delta^{(m)} f(x) \approx h^m f^m(x)$

and as error higher orders in D, this proves that $(m+1)$'th differences of an m'th degree polynomial vanish.

(c) One can use these formal method to derive some useful summation formulae.

For instance:

put $S' = \sum_{j=0}^{p-1} f(x+jh)$

$$= (1+E+\ldots+E^{p-1})\, f(x)$$

$$= \frac{(E^p-1)}{E-1}\, f(x).$$

now $\frac{x}{e^x-1} = \sum_{n=0}^{\infty} B_n\, x^n/n!$ defines Bernoulli numbers B_n (See page 48 of Mathews and Walker). Note $B_1 = -1/2$, $B_2 = 1/6$ etc.

So $S' = \frac{E^p-1}{hD} * \frac{hD}{e^{hD}-1}\, f(x)$

$$= \frac{E^p-1}{hD} \sum_{n=0}^{\infty} B_n\, (hD)^n/n!\; f(x)$$

$$= 1/h \int_x^{x+ph} f(y)\,dy + \sum_{n=1}^{\infty} B_n\, h^{n-1}/n! \left\{ f^{(n-1)}(x+ph) - f^{(n-1)}(x) \right\}$$

which is the Euler-Maclaurin summation formula (Mathews and Walker - page 365). It relates S' and integral $1/h \int_x^{x+ph} f(y)\,dy$ with error in form of derivatives. One can get error in form of differences by using

$$\frac{hD}{e^{hD}-1} = \frac{2\sinh^{-1} \tfrac{1}{2}\delta}{\tfrac{1}{2}\delta^2 + \delta\sqrt{1+\delta^2/4}}$$

getting denominator from $e^{hD}-1 = \epsilon - 1$

and $\delta^2 = (E^{1/2} - \epsilon^{-1/2})^2 = E - 2 + 1/E$

or $E^2 - (2+\delta^2)E + 1 = 0$

$$E = 1 + \delta^2/2 \pm \sqrt{(1+\delta^2/2)^2 - 1}$$

use + sign to get $E-1 \approx \delta$

or $E = 1 + \tfrac{1}{2}\delta^2 + \delta\sqrt{1+\delta^2/4}$.

In the problem set, we show how Euler's transformation (Mathews and Walker – page 54) can be derived by using similar formal manipulation.

In the previous pages, we interpreted

$$\frac{1}{D} f(x) \quad \text{as} \quad \int^{x} f(y)\,dy$$

This is correct as

$$D\left[g(x) = \left(\tfrac{1}{D} f(x)\right)\right] = \left(D \cdot \tfrac{1}{D}\right) f(x) = f(x)$$

and $\int^{x} f(y)\,dy$ is the correct Solution of this differential equation

This Solution is undefined upto a constant, but this is irrelevant as operator $E^{p} - 1$ annihilates a constant.

Ph 129c

Lecture May 19, 1988

Interpolation and Function Approximation

Newton's Formula - equally spaced grid points

Lagrange - general grid positions but preordained

Chebyshev - choose grid positions to minimize error in a region

Analyticity (Transformation of Variables)

Experimental Errors

References are:

General: Mathews and Walker, chapter 13.
(as usual, reasonable introduction to subject)

M. Abramowitz and I. Stegun: "Handbook of Mathematical Functions": this book has already been recommended for theory of special functions; it is also vital when calculating same as it has both formulae and actual values to check your subroutine with.

B. Noble: "Numerical methods" (Oliver and Boyd)

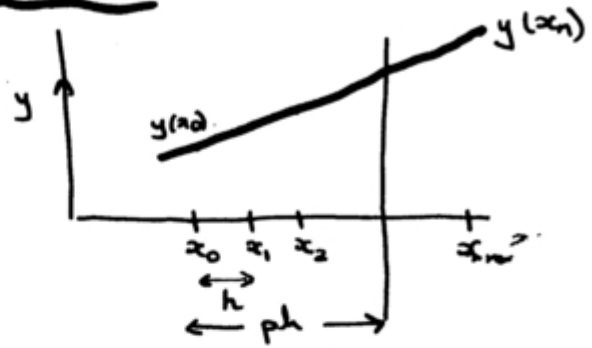
"oldie but goodie"

Caltech

E. Isaacson and H.B. Keller: "Analysis of Numerical methods": this book is very rigorous but still is quite readable.

E4/2 Equidistant Interpolation Formula

(i) Newton's Formula



Suppose we are given $n+1$ equidistant points x_m ($x_m = x_0 + mh$, $m = 0, 1 \ldots n$) and we know $y_m = f(x_m)$, what is our best estimate of $y_p = y(x_0 + ph)$?

$$y_p = E^p y_0 = (1+\Delta)^p y_0$$

$$= y_0 + p\,\Delta y_0 + p(p-1)\,\Delta^2 y_0 / 2! + \ldots \quad (E4.1)$$

which is still exact. However the values y_m ($0 \le m \le n$) are sufficient to calculate the first n differences $y_0 \ldots \Delta^n y_0$. If we truncate (E4.1) after $\Delta^n y_0$ term, then we get an approximate formula for y_p known as Newton's interpolation formula.

As the truncation $\Delta^m y = 0$, $m \geq n+1$ would

be exact for a polynomial of degree n, it is clear that Newton's formula is entirely equivalent to taking for $f(x_p)$ the value of the unique nth order polynomial approximation to $f(x)$ that coincides with the function at the $n+1$ points $x_0 \ldots x_n$.

(ii) Everett's Formula

Now it is clear intuitively that such interpolation is most appropriate for values of $p \approx \frac{1}{2}n$ near the midpoint of the range of ~~the~~ x-values that $f(x)$ is known. (For $p \approx 0$, ~~it~~ the tail is wagging the dog). However the formula (E4.1) is not very convenient for such large $p \approx \frac{1}{2}n$ near the optimal value. However we can <u>rearrange</u> (E4.1) by using central δ and not forward Δ differences.

i.e. use $\delta = E^{1/2} - E^{-1/2} = 2\sinh A/2$ where $E = e^A$, $A = hD$

To get calculable powers of δ (i.e.

even powers) we assume an even number of grid points which we label

$$x_{-n} \; x_{-n+1} \; \cdots \; x_0 \; x_1 \; \cdots \; x_n \; x_{n+1}$$

and after some algebra (use $E = e^A$, $A = \frac{1}{2}\sinh^{-1}\frac{1}{2}\delta$ and express y_p.

$$y_p = \frac{\sinh pA}{\sinh A} y_1 + \frac{\sinh qA}{\sinh A} y_0 \;, \; q = 1-p.$$

Then write $\sinh[2p \cdot A/2]$ in terms of power series in $\sinh A/2$)

we find a formula of form

$$\begin{aligned} y_p = \; & (1-p)\, y_0 + p\, y_1 \leftarrow \text{usual linear interpolation.} \\ & + E_2 \delta^2 y_0 + E_4 \delta^4 y_0 + \cdots \\ & + F_2 \delta^2 y_1 + F_4 \delta^4 y_1 + \cdots \end{aligned} \qquad (E4.2)$$

where E and F are called Everett coefficients. Explicitly

$$E_{2n} = \binom{1-p+n}{2n+1} \qquad F_{2n} = \binom{p+n}{2n+1}.$$

Just like Newtons formula, (E4.2) truncated after δ^n terms is equivalent to fitting a $(2n+1)$-th degree polynomial through $x_n \cdots x_{n+1}$.

E4/3

General Polynomial Interpolation

(i) We now generalize the above formulae to non equidistant points $x_0 \ldots x_n$. So we have $n+1$ arbitrary x values $x_0 \ldots x_n$ and $n+1$ values $y_m = f(x_m)$ and wish to estimate $f(x)$ for some x value. We construct the unique n'th order polynomial $P_n(x)$ which satisfies

$$P_n(x_m) = y_m \qquad m = 0 \ldots n \qquad (E4.3)$$

The polynomial P_n has $n+1$ coefficients a_i which are to be determined by the $n+1$ conditions (E4.3). So the problem looks soluble in principle. In fact we can write down the answer explicitly as:-

$$P_n(x) = \sum_{k=0}^{n} L_k(x)\, y_k$$

$$L_k(x) = q_k(x) \Big/ q_k(x_k) \qquad (E4.4)$$

$$q_k(x) = (x-x_0) \ldots (x-x_{k-1})(x-x_{k+1}) \ldots (x-x_n).$$

(E4.4) clearly satisfies (E4.3) and is known as Lagrange's interpolation

formula. If x_m were equidistant, $x_m = x_0 + mh$, then Lagrange is just a rearrangement of Newton's or Everett's formula in terms of function values - not differences

(ii) Error - Analysis

The analysis of the x dependence of the truncation error in (E4.4) is rather neat. First we remind you of:

Rolle's Theorem: If $g(x)$ is a suitably nice function of x in $[a,b]$ which satisfies $g(a) = g(b)$, then there is a ξ in $[a,b]$ such that $g'(\xi) = 0$.

Define $p_{n+1}(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$ and as the error in (E4.4) is clearly zero at $x_0\, x_1 \ldots x_n$, we write

$$f(x) = P_n(x) + p_{n+1}(x)\, S(x)$$

and try to find $S(x)$.

Define R as the range containing

$x_0 \ldots x_n$ and x. (We cover extrapolation and so R may be bigger than the range $x_0 \ldots x_n$). Fix x, and let X vary over R and put

$$F(X) = f(X) - P_n(X) - p_{n+1}(X)\, S(x)$$

Then $F(X)=0$ at $n+2$ values of X

i.e. $X = x$ and $x_0, x_1 \ldots x_n$

Apply Rolle's theorem between each pair of zeros of $F(X)$: we deduce that $F'(X)=0$ at $n+1$ values of X in R. We can continue process, showing recursively that $F^{(n+1)}(X)$ vanishes at least once in R. Let this be at $X = \xi$.

$$0 = F^{(n+1)}(\xi) \equiv f^{(n+1)}(\xi) - \underset{\text{polynomial of degree } n.}{0} - (n+1)!\, \underset{\text{fixed.}}{S(x)}$$

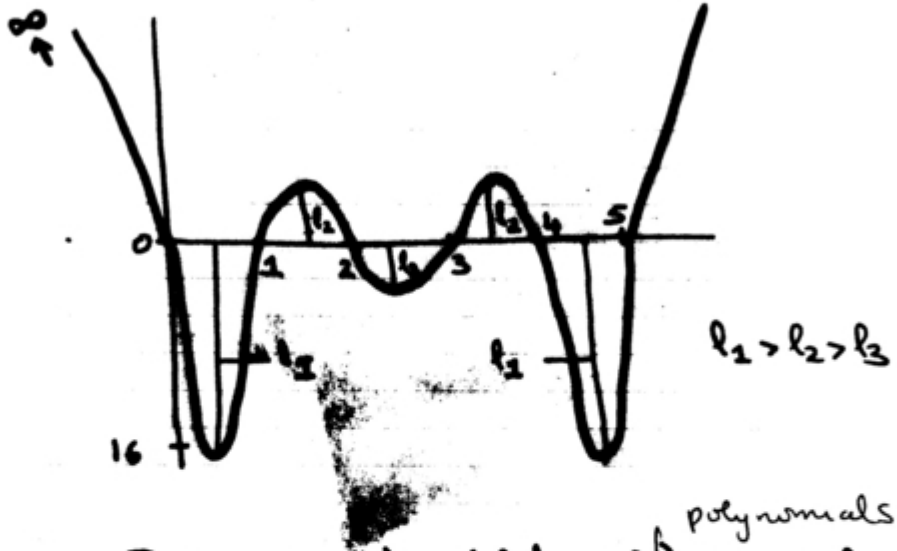
<u>or</u> $$S(x) = f^{(n+1)}(\xi)/(n+1)!$$

and total error is $p_{n+1}(x)\, f^{(n+1)}(\xi)/(n+1)!$

ξ is implicitly a function of x but this formula indicates that error is bounded

by $C\, p_{n+1}(x)$ and so controlled by $p_{n+1}(x)$.

For equidistant $x_m = m$ and $n = 5$



This $p_{n+1}(x)$ (like all polynomials) rapidly grows as x venturs outside $0 \to 5$: thus extrapolation is difficult*. However the error is clearly smallest in [2, 3] of intervals in 0 to 5. This a manifestation of our comment that Newton's formula was best for p in middle of range. However we will now discuss a choice of x_i such that the l_i are all of equal modulus and interpolation is uniformly good (bad) throughout range,

* note chebyshev are best for extrapolation - so is best for [illegible] Miller p. 270.

E4/4 Chebyshev Series

(i) Interpolation assumes that function given at ordained points outside our control. We now consider what happens if we are also allowed to choose the positions of the points.

Definition: Let $f(x)$ be a function to be approximated in $x \in [a,b]$ by a polynomial $p_n(x)$ of degree n. Then the best polynomial approximation to $f(x)$ is such that $\max_{x \in [a,b]} |f(x) - p_n(x)|$ is as small as possible. This called the minimax principle.

One can prove that if $f(x)$ is continuous, then $p_n(x)$ exists satisfying the following property:

Theorem: The error $\epsilon_n(x) = f(x) - p_n(x)$ has property that it attains its maximum modulus $|\epsilon_n(x)|$ value $= L$ at, at least, $n+2$ distinct points in $[a,b]$ and is alternately positive and negative there

Note 1 or 2 of these points could be end points $x=a$ or $x=b$.

Proof : we cannot do this rigorously. We must assume that there does exist a $p_n(x)$ satisfying conditions of theorem. This is not unreasonable as we have a grand total of $2(n+2)$ conditions (derivative $=0$, value $=\pm L$ at $n+2$ points) on $2(n+2)$ parameters ($n+2$ x-values, one L value, $n+1$ coefficients of polynomial).

Then suppose theorem is not true and the best polynomial $q_n(x) \neq p_n(x)$. Then let $\epsilon'_n(x) = f(x) - q_n(x)$ be error and trivially $|\epsilon'_n(x)| < L$ for all L.

Consider $r(x) = \epsilon_n(x) - \epsilon'_n(x) = q_n(x) - p_n(x)$: at x values where $\epsilon_n(x) = \pm L$, $r(x)$ has same sign as $\epsilon_n(x)$ i.e. $r(x)$ has $n+1$ sign changes or zeroes. But $r(x)$ is a polynomial of degree n and this is impossible. Q.E.D.

Isaacson and Keller cover the theory of best polynomials rigorously : here we

leave the general theory and just show why Chelyshev polynomials are sometimes good approximations to the best polynomial.

(a) Chelyshev Polynomials

(a) Mathematically, the Chelyshev polynomial is defined by

$$T_r(x) = \cos(r \cos^{-1} x)$$

as a polynomial of degree r in x.

The first few polynomials are

$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_2(x) = 2x^2 - 1$$
$$T_3(x) = 4x^3 - 3x \quad \ldots$$

They are examples of orthogonal polynomials we studied earlier this year. The orthogonality relation is:

$$\int_{-1}^{+1} T_r(x)\, T_s(x)\, dx / \sqrt{1-x^2} = \pi \quad r = s = 0$$
$$= \tfrac{1}{2}\pi \quad r = s \neq 0$$
$$= 0 \quad r \neq s$$

as is obvious transforming to $x = \cos\theta$

We expand functions

$$f(x) = \sum_{r=0}^{\infty}{}' a_r T_r(x)$$

where $a_r = \frac{2}{\pi} \int_{-1}^{+1} f(x) T_r(x) / \sqrt{1-x^2}$

and $'$ on Σ' indicates we should halve first term to account for anomaly in orthogonality condition.

Note $\partial T_r / \partial x = -\sin(r \cos^{-1} x) \, \partial / \partial x \, [\cos^{-1} x]$

$$= \frac{1}{\sqrt{1-x^2}} \sin[r \cos^{-1} x]$$

$$= 0 \quad \text{if} \quad r \cos^{-1} x = s\pi$$

($s = 1 \ldots r-1$ * $s = 0, r$ illegal due to $1/\sqrt{1-x^2}$).

(b) So maxima and minima of $T_r(x)$ occur for $x = \cos[s\pi/r]$ and value ~~at~~ these points is $\cos(s\pi)$ i.e. $(-1)^s$.

meanwhile, derivative doesn't vanish there, but $T_r(x) = 1$ at $x = 1$ and $(-1)^r$ at $x = -1$. Thus $T_r(x)$ has ~~are~~ $r+1$ maxima and minima in $[-1, 1]$ and is alternately ± 1 at these points. Now we can easily prove:

Theorem: If $f(x)$ is a polynomial of degree $n+1$, then $\sum_{r=0}^{n}{}' a_r T_r(x)$ is the best polynomial approximation of degree n in $[-1,+1]$.

Proof: $$f(x) = \sum_{r=0}^{n+1}{}' a_r T_r(x) \text{ exactly}$$
$$= p_n(x) + a_{n+1} T_{n+1}(x)$$
where $p_n(x) = \sum_{r=0}^{n}{}' a_r T_r(x)$.

So $\epsilon_n(x) = f(x) - p_n(x)$ satisfies condition of our earlier theorem on p. 60 and so $p_n(x)$ is best polynomial

Q.E.D.

The above shows that, for arbitrary $f(x)$, if error dominated by next term in series then Chebyshev expansion will be <u>near</u> best polynomial approximation. Of course, Chebyshev expansion is much easier to find than true best expansion which is why its approximate bestness is of interest.

We can see the same effect in

the error analysis of sect. 4/3. We found error $s(x)(\equiv \epsilon_n(x)) = \hat{p}_{n+1}(x)\, f^{n+1}(\xi)$ where $\hat{p}_{n+1}(x)$ is polynomial vanishing at $n+1$ points where $f(x) = p_n(x)$, the interpolating polynomial. Choosing the $n+1$ points as the zeroes of $T_{n+1}(x)$ gives $\hat{p}_{n+1}(x) \propto T_{n+1}(x)$ and the error is uniform:  not  of equidistant formula.

Again this is only precise if $f^{n+1}(\xi)$ is constant i.e. $f(x)$ is a polynomial and we're back with the theorem on page 64.

(c) We quote a rather pretty theorem to be found on page 32 of Snyder. Thus Chebyshev polynomials give best ~~poly~~ convergence of all ultraspherical polynomials. Put expansion polynomial

$$e_n(x) = c_n\,(1-x^2)^{-\alpha}\, \frac{d^n}{dx^n}\,[1-x^2]^{n+\alpha}$$

where $\alpha = \infty$ is Taylor series $e_n(x) \propto x^n$.

$\alpha = -1/2$ is Chebyshev.

$\alpha = 0$ is Legendre.

Then expand arbitrary $f(x)$

$$f(x) = \sum_{r=0}^{n} a_r e_r(x) + \epsilon_n$$

Synder shows that

$$|\epsilon_n(x)| \leq f^{(n+1)}(\xi) \Big/ \beta_n (n+1)!$$

$\beta_n = 1$: Taylor WORST

$= 2^n / \sqrt{\pi(n+1)}$: Legendre

$= 2^n$: Chebyshev BEST

E4/5 Analyticity

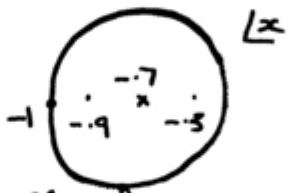
Previously we indicated that choice of interpolation points and expansion polynomials was not an art but had a respectable theory behind it. We now overthrow the last of "artistic" assumptions i.e. why use a polynomial?

The answer is that, in general, there is no reason and there are better expansion functions. However, unlike Chebyshev series which are famous in numerical analysis, this point seems not to be covered in the standard books on numerical analysis. In fact, it is a classical mathematical theory with rigorous theorems and probably should be used more. It is just being discovered in high-energy physics but hasn't had fantastic success yet.

I will indicate method by a trivial example.

Suppose we wish to approximate $f(x) = 1/(1+x)$ over the range $x = -.9$ to $-.5$. Polynomials are possible but quite a few will be needed as, for instance, Taylor Series about $x = -.7$ has a radius of convergence 0.3 i.e in

$$f(x) = \sum_n a_n (x+.7)^n \quad , a_n \sim (1/.3)^n.$$



On the other hand, if we tried to expand $1/(2+x)$ in same range we would find faster convergence $a_n \sim (1/1.3)^n$.

These examples indicate that for analytic functions, the position of singularities in complex plane control convergence of polynomial expansion and hence accuracy of series truncated after a given term. Now conformal maps can move singularities and hence improve convergence.

For instance to approximate $f(x) = 1/(1+x)$ we move singularity at $x = -1$ to ∞ by transformation $w = 1/(1+x)$

$\Rightarrow$ $f(x)$ becomes $f(w) = w$

and polynomials in w certainly converge fast (note these are not polynomials in z).

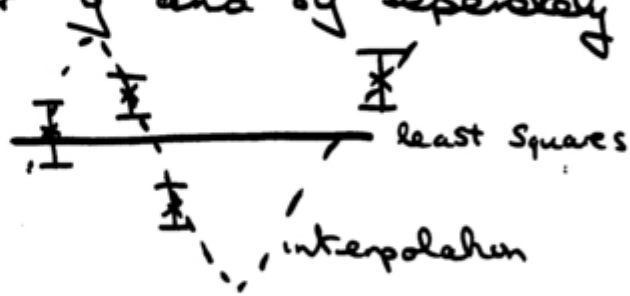
So theory is: given domain of analyticity i.e. singularity structure and domain D' of approximation: map former to achieve fastest possible rate of convergence of polynomial approximations in D'. There is a unique answer for given D and D'.

Warning: in practice, the behaviour at ∞ of the original function is of importance. This point gets mapped to finite w and can produce a nasty (essential) singularity.

E4/5 Statistical Warning.

The techniques we were discussing in the previous sections are only appropiate for functions $f(x)$ known with neglectable error. Suppose $f(x)$ is measured experimentally to be $y_i \pm \delta y_i$ at $x = x_i$ $(i = 0 \ldots n)$. If you wish to find $f(x)$ in $[x_0\ x_n]$, one should not use E4/1..4 for y and δy seperately

e.g.



is consistent with constant $f(x)$ and the third degree (as 4 points) interpolating polynomial is ridiculous. For such problems, one should least squares techniques discussed in statistics.

i.e. minimize $\sum_{i=0}^{n} \left[\frac{P_m(x_i) - y_i}{\delta y_i} \right]^2$

E5 Numerical Integration

The problem of finding $I = \int_a^b f(x) g(x)\, dx$ is closely related to that of expanding $f(x)$ in $[a,b]$. If the expansion functions are nice, we can plug in expansion for f and do term by term integral. It divides into

E5/1,2 $f(x)$ assumed known at prescribed points. (cf. Lagrange interpolation)

E5/3: we can choose points where $f(x)$ to be calculated (cf. Chebyshev expansion)

E5/4: multidimensional integrals.

It is treated in Mathews and Walker, p. 349-352 and Noble chapter IX. Also Abramowitz and Stegun have a great compendium of rules.

E5/1: Newton-Cotes Formulae

(i) We write

$$I = \int_a^b f(x)\, g(x)\, dx \qquad (E5.1)$$

where $f(x)$ is assumed known at $x_0 \ldots x_n$ and $g(x)$ is sufficiently simple so that integrals (E5.3), involving it, can be calculated. of course, we split off $g(x) \neq 1$ to make $f(x)$ smoother and more reasonably approximated as a polynomial. we assume the form:

$$I = \sum_{r=0}^{n} w_r f(x_r) \qquad \text{(E5.2)}$$

where the $n+1$ w_r's are to be found by requiring I to be exact for all polynomials of degree $\leq n$. This is easy - for put

$$I_m = \underbrace{\int_a^b x^m g(x)\,dx}_{\text{known \#'s}} \qquad \text{(E5.3)}$$

and our condition becomes:

$$I_m = \sum_{r=0}^{n} w_r x_r^m \quad : 0 \leq m \leq n \qquad \text{(E5.4)}$$

(linear nature of integral ensures that (E5.4) guarantees exactness of (E5.2) for all polynomials of degree $\leq n$). we can write (E5.4) as:

$$A \underline{w} = \underset{\sim}{b} \qquad \text{(E5.5)}$$

$\underline{w}$ is column vector of w_i's, $\underline{b}$ is column vector of I_m's and A is $(n+1)\times(n+1)$ matrix of coefficients $A_{mr} = (x_r)^m$. It is easy to show that A is nonsingular (if x_r distinct) and hence that (E5.5) can be solved and (E5.2) established for any $g(x)$ in (E5.1).

Usually (E5.5) is solved – once and for all for given $g(x)$ – and then (E5.2) can be used quickly for any $f(x)$ that crops up. Examples:

(a) Trapezoidal Rule

$n=1$, $g(x)=1$ $\quad x_0 = a$, $x_1 = b$.

$$\int_a^b f(x)\,dx \approx \tfrac{1}{2}h\,[f(a) + f(b)] \qquad h = b-a \qquad \text{(E5.6)}$$

(b) Simpson's Rule:

$n=2$, $g(x)=1$, $x_0 = a$, $x_1 = \tfrac{1}{2}(a+b)$, $x_2 = b$

$$\int_a^b f(x)\,dx = \tfrac{1}{3}h\,[f(x_0) + 4f(x_1) + f(x_2)] \qquad \text{(E5.7)}$$

where $h = \tfrac{1}{2}(b-a)$

Both these are equi-spaced (in x_i) rules with $g(x)=1$ and the generalization

to arbitrary n are called Newton-Cotes formulae. (See Abramowitz and Stegun to find them explicitly written out). We can relax restrictions:

(a) x_i equidistant
(b) $x_0 = a$ and/or $x_n = b$
(c) $g(x) = 1$

The reader need only solve (E5.5) to find relevant formulae for any n.

(ii) The errors in (E5.6,7) are written in the form:

trapezoidal - (E5.6) — $-h^3 f''(\xi)/12$ (E5.8)

Simpson - (E5.7) — $-h^5 f^{(iv)}(\xi)/90$.

where, as usual, ξ is some value of x in $[a,b]$.

Note that, as expected, trapezoidal formula is exact for linear functions (error $\propto f''(\xi)$ vanishes for $f(x) = \alpha x + \beta$) but Simpson's rule - which as derived is exact for quadratic $f(x)$ - is unexpectedly exact for cubics. In other words, its error is small and it is a widely used formula

in its extended form:

$$\int_{x_0}^{x_{2n}} f(x)\,dx = \frac{h}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + \cdots + f_{2n} \right] - \frac{nh^5}{90} f^{(iv)}(\xi) \qquad \text{(E5.9)}$$

where $f_i = f(x_i)$ and x_i are equidistant. Get this by applying Simpson to intervals x_0 to x_2 x_2 to x_4 .. x_{2n-2} to x_{2n} in succession. One could also generate a formula in terms of the ~~Newton Cotes~~ same function values $f(x_i)$ by using the Newton Cotes formula appropiate to $2n+1$ points. This gains an extra derivative in error just like Simpson, and error is $c h^{2n+3} f^{(2n+2)}(\xi)$ where c is some known number. In practice this is not often used because

(a) The implied interpolation of $f(x)$ over a wide range by a high order polynomial can produce ~~an~~ disastrous results when $f(x)$ not polynomial. (eg.

has singularities) On the other hand, local interpolation by $f(x)$ by a low order polynomial (used in (E5.9)) is much safer

(b) It is much easier to estimate error in (E5.9) as we describe in Sect. E5/2.

(iii) Points of Technique

Note, when evaluating $I = \int_a^b f(x) g(x) dx$, one can choose between

(a) Change of variable e.g. $dy = g(x)\, dx$ converts integral to $g(y) = 1$ form.

(b) Newton-Cotes (or Gauss - see E5/3) formula with weight function $g(x)$.

This remarks also hold for change of integration range (e.g. to take b from ∞ to finite value)

In each case, one must simply use criterion.

Is $f(x)$ or $f(y)$ better approx ~~more accurately~~ indicated by a polynomial?

One can use same criterion to

both choose $g(x)$ and/or subtract off some particularly nasty part (e.g. singularity) of $f(x)$.

e.g. $$I = \int_a^b \overset{= f_1(x) + f_2(x)}{f(x)}\, g(x)\, dx = I_1 + I_2$$

$$I_1 = \int_a^b \underset{\text{polynomial}}{f_1(x)}\, g(x)\, dx \quad \text{done numerically}$$

$$I_2 = \int_a^b \underset{\text{evil function.}}{f_2(x)}\, g(x)\, dx \quad \text{done analytically}$$

E5/2 Romberg Integration

A very convenient way of evaluating integrals is to repeatedly apply Simpson's rule – halving interval each time

```
a                       b
x           x           x        Step 1

•     x     •     x     •        2

•  x  •  x  •  x  •  x  •        3
         →x
```

The point is that, as diagram indicates, one need only evaluate half as many (x in diagram) functions as you expect because you use old values (• in figure)

As the error is proportional to nh^5 and n doubles and h halves each go, error should go down by $\approx 1/16$ each iteration and convergence is fast. So, in practice, one halves h until change in integral between steps is less than some assigned value.

There is a nice discussion of this in Noble §9.3 p. 231 onwards (Vol. II).

E5/3 Gaussian Integration

We are evaluating the same formula

$$I = \int_a^b g(x) f(x)\, dx \qquad (E5.1)$$

in terms of the same ansatz

$$I = \sum_{r=0}^{n} w_r f(x_r) \qquad (E5.2)$$

Now we choose <u>both</u> w_r and x_r (a grand total of $2n+2$ parameters)

to make (E5.2) exact for polynomials of degree $\leq 2n+1$. ($2n+2$ equations analogous to (E5.3)).

Let $P_n(x)$ be the polynomials orthogonal wrt weight function $g(x)$

$$\int_a^b g(x)\, P_n(x)\, P_m(x)\, dx = h_n \delta_{nm} \qquad \text{(E5-10)}$$

Let $x_i^{(n)}$ be the ith zero of $P_n(x)$

Define $K_n(x,y)$ by the formula

$$K_n(x,y) = \sum_{k=0}^{n} P_k(x)\, P_k(y) / h_k \qquad \text{(E5-11)}$$

If $\pi_n(x)$ is any polynomial of degree n, then one can easily show:

$$\int_a^b g(x)\, K_n(x,y)\, \pi_n(x)\, dx = \pi_n(y) \qquad \text{(E5-12)}$$

It is not difficult to see that (E5.2) is satisfied by

$$x_i = x_i^{(n+1)}$$
$$w_i = 1 / K_{n+1}(x_i^{(n+1)}, x_i^{(n+1)}). \qquad \text{(E5-13)}$$

Take as a complete set of polynomials of degree $2n+1$

a) $P_{n+1}(x) \cdot \left.\begin{matrix} 1 \\ x \\ x^2 \\ \vdots \\ x^n \end{matrix}\right\}$ I is zero and is ensured by $x_i = x_i^{(n+1)}$.

b) $K_{n+1}\left(x_i^{(n+1)}, x\right) \cdot (x - x_0^{(n+1)})(x - x_1^{(n+1)}) \ldots \underset{\text{omit } (x - x_i^{(n+1)})}{\uparrow} (x - x_n^{(n+1)})$

By construction, the only term that contributes to I for this polynomial is that with $r = i$ and this gives formula for w_i.

For $g(x)=1$, we get normal Gauss formulae with:

$$x_i = i\text{-th zero of } P_{n+1}(x)$$
$$w_i = \frac{2}{[1-x_i^2]} [P'_{n+1}(x_i)]^{-2} \quad \text{(E 5-14)}$$

Clearly Gaussian integration is best when $f(x)$ is very hard to evaluate and one must minimize value of n. However it is very hard to estimate error in calculation as the simple iterative scheme described in E5/2 for Simpson's rule cannot work. Not only is hard to evaluate weights and x_i's as n tends $\to \infty$, but you have to recalculate all n functions each time.

There is no overlap in x_i between different n values. (Note Gauss only saves factor of 2 in further evaluations and Romberg is simpler to code and saves same factor of 2 if you have to iterate).

So use Romberg if you have a few integrals which are needed to prescribed accuracy. Use Gaussian integration, when you have a lot of integrals with similar $f(x)$ and the right value of n can be determined before hand.

As the error is proportional to nh^5 and n doubles and h halves each go, error should go down by $\approx 1/16$ each iteration and convergence is fast. So, in practice, one halves h until change in integral between steps is less than some assigned value.

There is a nice discussion of this in Noble §9.3 p.231 onwards (Vol. II).

E5/3 Gaussian Integration

We have already discussed this under orthogonal polynomials and so we won't do all the details.

We are evaluating the same formula

$$I = \int_a^b g(x) f(x)\, dx \qquad (E5.1)$$

in terms of the same ansatz

$$I = \sum_{r=0}^{n} w_r f(x_r) \qquad (E5.2)$$

Now we choose <u>both</u> w_r and x_r (a grand total of $2n+2$ parameters)

to make (E5.2) exact for polynomials of degree $\leq 2n+1$. ($2n+2$ equations analogous to (E5.3)).

You remember that the problem was solved in terms of polynomials orthogonal wrt weight function $g(x)$. For $g(x)=1$, we get normal Gauss formulae with:

$$x_i = i\text{-th zero of } P_{n+1}(x)$$
$$w_i = \frac{2}{[1-x_i^2]} [P'_{n+1}(x_i)]^{-2} \qquad \text{(E5.10)}$$

Clearly Gaussian integration is best when $f(x)$ is very hard to evaluate and one must minimize value of n. However it is very hard to estimate error in calculation as the simple iterative scheme described in E5/2 for Simpson's rule cannot work. Not only is hard to evaluate weights and x_i's as n iterates $\to \infty$, but you have to recalculate all n functions each time.

There is no overlap in x_i between different n values. (Note Gauss only saves factor of 2 in further evaluations and Romberg is simpler to code and saves same factor of 2 if you have to iterate).

So use Romberg if you have a few integrals which are needed to prescribed accuracy. Use Gaussian integration, when you have a lot of integrals with similar $f(x)$ and the right value of n can be determined before hand.

E5/4 Monte-Carlo Integration

(i) The general integration formula can be written as:

$$I = \int_a^b f(x)\,dx = \sum_{i=0}^{n} f(x_i)\,w_i$$

where we temporarily take $g(x)=1$.

The n-points x_i and weights w_i are determined

E6 Ordinary Differential Equations

The discussion in Mathews and Walker, page 353-355, is quite good. However, as we take relish in later, they perversely describe a very poor version of Predictor-Corrector method. As usual Noble is good and more complete while still being elementary. (chapter 10). My bible, Modern Computing methods, is this time rather poor.

Remember that ordinary differential equations can be divided by their boundary conditions

A ———— B

e.g. $y'' = f(x, y)$

y, y' given at A is initial value problem

y given at A and B is a boundary value problem.

The first two sections cover initial value problems. Note that by putting

Note: ignore references
"Numerical Recipes is better
than older cited references"

Ph 129c

May 24 and 25, 1988

Numerical Integration

Newton-Cotes

Trapezoidal
Simpson's rules

Romberg Integration (iterated Simpson)

Gaussian Integration

[Monte-Carlo methods] - Discussed earlier

Ordinary Differential Equations

Initial Value Problems

Runge-Kutta

Predictor-Corrector

Error Analysis

Boundary Value Problems

Partial Differential Equations

Elliptic
Parabolic equations
Hyperbolic

E7: Partial Differential Equations

These are not subject to very different numerical techniques from ordinary differential equations. We first remember the theory of characteristics (Mathews and Walker, pages 217→226; Modern Computing Methods, pages 101→104). This classified equations into

Elliptic : no real characteristics
Parabolic : two degenerate characteristics
Hyperbolic : two distinct " " .

The first two are particularly similar to ODE. Elliptic equations typically give boundary value problems and one gets matrix inversion/eigenvalues formulation after making finite difference approximation to, say, ∇^2. A parabolic equation e.g. $\partial^2\phi/\partial x^2 = \partial\phi/\partial t$ is

typically a mixture of matrix (in x) and recursion (in t) techniques. For instance one can take a grid in x ($x_0 \ldots x_{n+1}$) and reduce PDE to a coupled set of ODEs for $\frac{d}{dt}\psi(x_n, t)$ where you make standard finite difference approximation for $\frac{\partial^2 \psi}{\partial x^2}$.

If h is x-grid size, these read:

$$h^2 \, d\psi_1/dt + 2\psi_1 - \psi_2 = \psi_0 \quad \text{known for all } t$$

$$h^2 \, d\psi_2/dt - \psi_1 + 2\psi_2 - \psi_3 = 0$$

$$\vdots$$

$$h^2 \, d\psi_n/dt - \psi_{n-1} + 2\psi_n = \psi_{n+1}$$

where $\psi_n(t) = \psi(x_n, t)$.

Of course, boundary conditions give $\psi_n(0)$ and we can use E6/1,2 to solve above.

More distinctive are hyperbolic equations.

E7/1 Hyperbolic Equations

(i) To recapitulate characteristics:

equation is $a\dfrac{\partial^2 u}{\partial x^2} + b\dfrac{\partial^2 u}{\partial x \partial y} + c\dfrac{\partial^2 u}{\partial y^2} = e$ (E7.1)

define $p = \partial u/\partial x \qquad q = \partial u/\partial y$

$r = \partial^2 u/\partial x^2 \qquad s = \dfrac{\partial^2 u}{\partial x \partial y} \qquad t = \partial^2 u/\partial y^2$

Differentials

$$du = p\,dx + q\,dy \qquad (E7.2)$$

$$dp = r\,dx + s\,dy \qquad (E7.3)$$

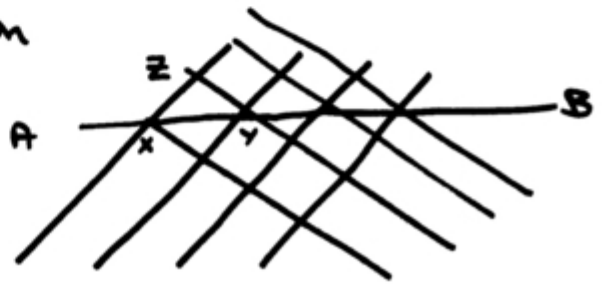
$$dq = s\,dx + t\,dy \qquad (E7.4)$$

Characteristics come from putting determinant of (E7.1,2,3,4) = zero

$$a(dy)^2 - b\,dy\,dx + c(dx)^2 = 0 \qquad (E7.5)$$

and are curves ~~where~~ on which Cauchy conditions (given u, p, q) do <u>not</u> specify solution

Suppose AB have given boundary conditions and



red lines are characteristics.

(iii) The method to be used, is most easily illustrated when we choose characteristics as axes. i.e. equation is:

$$b\,\partial^2 u/\partial x\,\partial y = e \qquad (E7.1')$$

characteristics $-b\,dy\,dx = 0$ i.e $x = \text{const.}$ or $y = \text{const.}$

Consider (E7.3)

$$dp = r\,dx + s\,dy$$
$$= r\,dx + e/b\,dy$$

So on line $x = \text{const}$

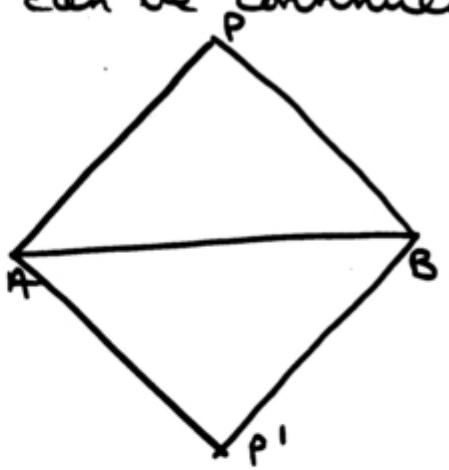
$$dp = e/b\,dy \qquad (E7.6)$$

Similarly on $y = \text{const.}$

$$dq = e/b\,dx \qquad (E7.7)$$

Thus, in previous diagram we know (u, p, q) at X and Y. Integrate (E7.6) along XZ (Suppose this is $x = \text{const}$) $\Rightarrow$ p at Z. Similarly integrate (E7.7) along YZ

to give q at Z. (E7.2) then gives u at Z. This process can be continued and clearly, given boundary conditions along AB, we can find u in all of APBP'.



(iii) Let's generalize method to case where characteristics are not orthogonal axes. Eliminate r and t between (E7.1) $(ar + bs + ct = e)$ (E7.2) and (E7.3)

$$a\,dy\,(E7.3) + c\,dx\,(E7.4) \equiv$$

$$a\,dp\,dy + c\,dq\,dx = e\,dx\,dy + s\left\{a(dy)^2 + c(dx)^2 - b\,dx\,dy\right\}$$

Now along a characteristic, coefficient of s is identically zero, and we get

$$a\,dp\,dy + c\,dq\,dx = e\,dx\,dy \qquad (E7.8)$$

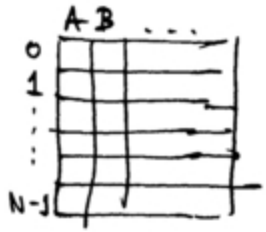
(where dx/dy satisfies (E7.5)).

(E7.8) is an ordinary differential equation and implies a functional relation between p and q along characteristics. So, as before, integrate (E7.8) along XZ to give one relation between p and q at Z, integrate (E7.8) along YZ to find another relation. Combine them to find p and q at Z. The procedure is then as in simple example.

Hopfield & Tank, TSP:

TSP: given N cities A, B, ..., X, Y, .. and the distances T_{XY} find the shortest path visiting each city exactly once.

Neural variables: take N^2 neurons V_{Xi}, where X is the city label and i is the path index (i.e. i=0 for the first visited city, i=1 for the second etc.)



Constraints at the fixed point:

c) cost minimal (= path length)

s) syntax constraints:

s0) $\overset{*}{V}_{Xi} = 0, 1$ (definite decision)

s1) $\sum_X \overset{*}{V}_{Xi} = 1$ (salesman conservation)

s2) $\sum_i \overset{*}{V}_{Xi} = 1$ (~~once~~ one visit constraint)

H&T action:

$$A = \beta_c A_c + \sum \beta_{s_i} A_{s_i}$$

$$A_c = \sum_{XYi} T_{XY} V_{Xi} V_{X,i+1}$$

$$A_{s_0} = \sum_{Xi} V_{Xi}(1 - V_{Xi})$$

$$A_{s_1} = \sum_i \left(\sum_X V_{Xi} - 1\right)^2$$

$$A_{s_2} = \sum_X \left(\sum_i V_{Xi} - 1\right)^2$$

$H = H\{S\}$

$S = \pm 1$ or $2(\eta - 1/2)$ $\eta = 0$ or 1

Steepest descent

$$S_i \to -\text{sign}\,\frac{\partial H}{\partial S_i}$$

Monte Carlo

Note only one change

$$S_i \to -S_i \quad \text{if } S_i \frac{\partial H}{\partial S_i} > 0$$

$$\to -S_i \quad \text{with probability } e^{-\Delta_i/T}$$

$$\text{if } \Delta_i \geq 0$$

$$\Delta_i = H(-S_i) - H(S_i)$$

as $T \to 0$

$$S_i \to -S_i \quad \text{if } S_i \frac{\partial H}{\partial S_i} > 0$$

i.e. $\to -\text{Sign}\,\partial H/\partial S_i$

Neural Networks

$$du_i/dt = -u_i - \frac{1}{T}\partial H/\partial s_i$$

$$s_i = \tanh u_i$$

as $du_i/dt \to 0$

$$s_i \to -\tanh\left(\frac{1}{T}\partial H/\partial s_i\right)$$

Note s_i continuous
or discrete.