Numerical Analysis

James D Emery

3/1/2013

Contents

1 Polynomial Evaluation

Let a polynomial have the form

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_nx^n.$$

We shall find an economical way to evaluate the polynomial and its derivatives at a point c. If we divide by $x - c$ we obtain a polynomial $q^0(x)$ and a

remainder r_0. That is

$$p(x) = (x-c)q^0(x) + r_0$$

$$= (x-c)(b_1 + b_2x + b_3x^2 + ... + b_nx^{n-1}) + b_0$$

Therefore we may compute q^0 and r_0 as follows. We have

$$b_n = a_n,$$

$$b_{n-1} = a_{n-1} + cb_n,$$

$$b_{n-2} = a_{n-2} + cb_{n-1},$$

$$...$$

$$b_0 = a_0 + cb_1,$$

and

$$r_0 = b_0.$$

Now we repeat this division $n-1$ more times getting

$$q^0(x) = (x-c)q^1(x) + r_1,$$

$$q^1(x) = (x-c)q^2(x) + r_2,$$

$$...$$

$$q^{n-2}1(x) = (x-c)q^{n-1}(x) + r_{n-1}.$$

We have divided $p(x)$ by $x-c$, n times, so that

$$q^{n-1}(x) = a_n.$$

Then

$$p(x) = (x-c)q^0(x) + r_0$$

$$= (x-c)((x-c)q^1(x) + r_1) + r_0$$

$$= (x-c)^2q^1(x) + r_1(x-c) + r_0$$

$$...$$

$$= a_n(x-c)^n + r_{n-1}(x-c)^{n-1} + r_{n-2}(x-c)^{n-2} + ... + r_0.$$

Differentiating k times for $0 \geq k \geq n$, we have

$$p^{(k)}(c) = k! r_k,$$

where we take $r_n = a_n$. In particular the value of the polynomial at c is

$$p(c) = r_0.$$

The following program does this calculation.

```
c polykder.ftn 0 through k polynomial derivative values
c 2/25/13
       implicit real*8 (a-h,o-z)
       dimension a(100),f(100)
       n=4
       a(1)=5
       a(2)=4
       a(3)=3
       a(4)=2
       a(5)=1
       k=4
       x=3
       call polykder(a,n,k,x,f)
       do i=1,k+1
        v =f(i)
        write(*,'(a,i2,a,g15.8)')' D(',i-1,') = ',v
       end do
       end
c
c+ polykder zero through kth derivative values of a polynomial
      subroutine polykder(a,n,k,x,f)
       implicit real*8 (a-h,o-z)
c input:
c  a-polynomial coefficients: p(x)=a(1)+ a(2)x+...+a(n+1)x^n
c  n-degree of polynomial
c  k- compute 0 through kth derivatives, 0 <= k <= n
c  x-evaluation point
c output:
```

```
c f-0 thru kth derivatives at x: p(x)=f(1), p'(x)=f(2), p''(x)=f(3) ...
c method: repeated division by (x-c) ( see numanal.tex)
c edited 2/25/13, from polval 11/16/94
      dimension a(*),b(100),f(*)
      np1=n+1
      do  i=1,np1
       b(i)=a(i)
      end do
      b(np1+1) = 0.
      if(k .eq. n)then
       kk=n
      else
       kk=k+1
      end if
      do  i=1,kk
       if(i .eq. 1)then
        m=1
       else
        m=m*(i-1)
       endif
       do  j=np1,i,-1
        b(j)=b(j) + x*b(j+1)
       end do
       f(i) = m*b(i)
      end do
      if(k .eq. n)then
       f(n+1)=m*n*a(n+1)
      endif
      return
      end
```

We are computing the 0 through 4 derivative values for $x = 3$. And here is what the program produces:

```
D( 0) =   179.00000
D( 1) =   184.00000
D( 2) =   150.00000
```

```
D( 3) =   84.000000
D( 4) =   24.000000
```

And here are some Maple commands and results from performing the same calculation on the polynomial

$$5 + 4x + 3x^2 + 2x^3 + x^4,$$

evaluated at $x = 3$.

```
e:=5 + 4*x + 3*x^2 + 2*x^3 + x^4;
eval(e,x=3);
179
eval(diff(e,x),x=3);
184
eval(diff(e,x,x),x=3);
150
eval(diff(e,x,x,x),x=3);
84
eval(diff(e,x,x,x,x),x=3)
24
```

2 Roots of a Cubic Polynomial

First we transform a real root to the unit interval. Let the cubic equation be

$$x^3 + ax^2 + bx + c = 0.$$

Let β = -sign(c) and $x' = \beta x$. Then

$$x'^3 + a\beta x'^2 + bx' - |c| = 0.$$

Let $\alpha = \sqrt[3]{|c|}$ and $x'' = x'/\alpha$. Then

$$x''^3 + \frac{a\beta}{\alpha}x''^2 + \frac{b}{\alpha^2}x'' - 1 = 0.$$

Define f(x") by the right side of this equation. We have $f(0) = -1$ and $f(\infty) = \infty$ so there is a positive root. If $f(1) > 0$ then there is a root

between 0 and 1. If $f(1) < 0$ then there is a root greater than 1. In that case let $x''' = 1/x''$ and then

$$x'''^3 - \frac{b}{\alpha^2}x'''^2 - \frac{a\beta}{\alpha}x''' - 1 = 0$$

has a root between 0 and 1. A real root of the original cubic is

$$x_1 = \alpha\beta x_1''$$

or

$$x_1 = \frac{\alpha\beta}{x_1'''}.$$

3 Synthetic Division and Newton's Method

We may write a polynomial

$$p(x) = a_0 + a_1x + ... + a_nx^n$$

in nested form as

$$a_0 + x(a_1 + x(a_2 + x(a_3 + ... + x(a_{n-1} + xa_n))..).$$

To evaluate p at c we compute

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + cb_n$$

.....

$$b_k = a_k + cb_{k+1}$$

.....

$$b_0 = a_0 + cb_1.$$

Then $p(c) = b_0$. Let

$$q(x) = b_1 + b_2x + ... + b_nx^{n-1}.$$

Comparing like terms we see that

$$p(x) = (x - c)q(x) + b_0$$

We have

$$\frac{dp}{dx} = q(x) + (x - c)\frac{dq}{dx}$$

and

$$\frac{dp(c)}{dx} = q(c).$$

Then Newton's method for a root c of p(x) is

$$c_m \to c$$

$$c_{m+1} = c_m - \frac{p(c_m)}{dp(c_m)/dx} = c_m - \frac{b_0}{q(c_m)} = c_m - \frac{b_0}{b_1'}$$

where

$$b_n' = b_n, b_k' = b_k + c_m b_{k+1}, k = 1, ..., n - 1.$$

4 Tschebyscheff Approximation

Suppose we are given the cubic polynomial

$$p(x) = x^3 + Ax^2 + Bx - 1$$

defined on the unit interval. We want to find the best quadratic approximation to this cubic in the minimax sense. That is we want to minimize the maximum difference on the interval.

The Tschebyscheff polynomial of degree n is defined by

$$T_n(x) = \cos(n \cos^{-1}(x)) = 2^{n-1}x^n + ...$$

and has the amazing property that of all polynomials of degree n and leading coeffient 2^{n-1}, it has the smallest maximum absolute value on the interval $[-1, 1]$. This value is 1, which is attained in the interval $n + 1$ times. All the zeroes of this polynomial lie in $[-1, 1]$. For the proof of this, see *Interpolation and Approximation* by Philip J. Davis (Theorem 3.3.4 p. 62). The idea of the proof is to assume that a second polynomial attains a smaller maximum

magnitude in [-1,1]. Subtract it from $T_n(x)$, getting a difference polynomial of degree $n-1$. The $n+1$ extreme points of $T_n(x)$ are attained with alternating sign, that is for example with a positive extreme value, then a negative, then a positive and so on. It is then obvious from the graph of such a function that if one draws a second curve which never attains these extremum values, then it must intersect the first curve at least n times. These intersection points correspond exactly to zeroes of the difference between the two curves. But the difference is of degree $n-1$ and can not have n zeroes, so there does not exist such a polynomial. This proves the theorem.

Returning to the approximation problem, we transform the domain [-1,1] of

$$T_3(x) = 4x^3 - 3x$$

to the interval [0,1], using the transformation $x \mapsto (x+1)/2$. We find that the Tschebyscheff polynomial becomes

$$32x^3 - 48x^2 + 18x - 1.$$

Dividing by 32 we have

$$x^3 - \frac{3}{2}x^2 + \frac{9}{16}x - \frac{1}{32}.$$

This polynomial has the smallest maximum magnitude of any monic cubic polynomial on [0,1] (monic means that the high order coefficient is 1). The maximum magnitude attained is $\frac{1}{32}$.

Consider the original cubic $p(x)$ written as

$$p(x) = x^3 + q(x)$$

We replace the x^3 term by the negative of the quadratic part of the normalized Tschebyscheff polynomial, getting a quadratic

$$r(x) = \frac{3}{2}x^2 - \frac{9}{16}x + \frac{1}{32} + q(x).$$

Then

$$\|p(x) - r(x)\| = \|x^3 - \frac{3}{2}x^2 + \frac{9}{16}x - \frac{1}{32}\| = \frac{1}{32}.$$

Because $p(0) = -1$ we have $\|p(x)\| \geq 1$ so the relative error is less than $1/32$, which is about 3 percent. The explicit quadratic approximation is

$$r(x) = (A + \frac{3}{2})x^2 + (B - \frac{9}{16})x - (\frac{31}{32})$$

5 The Maximum Curvature of a Cubic Polynomial

We will start by locating the extreme points of the cubic. Let

$$f(x) = ax^3 + bx^2 + cx + d.$$

The second derivative is zero at the inflection point, which is

$$\alpha = -\frac{b}{3a}.$$

To eliminate the squared term, let $z = x - \alpha$, then

$$f(x) = h(z) = az^3 + c_z z + d_z,$$

where

$$c_z = c - \frac{b^2}{3a}.$$

The cubic is an odd function about the point (α, d_z). The second derivative is also an odd function, so the curvature of the cubic is an odd function about the point α. The curvature of f is

$$\kappa_f(x) = \frac{f''}{(1 + f'^2)^{3/2}} = \frac{6ax + 2b}{(1 + (3ax^2 + 2bx + c)^2)^{3/2}}.$$

The curvature is preserved by the translation and

$$\kappa(x) = \kappa_h(z) = \frac{h''}{(1 + h'^2)^{3/2}}.$$

κ' is zero at points of extreme curvature. At an extreme point we have

$$0 = \kappa_z'(1 + h'^2)^{5/2} = h'''(1 + h'^2) - 3h''^2 h'.$$

Let $q(z)$ be the right side of this equation. We find that

$$q(z) = Az^4 + Bz^2 + C = 0,$$

where

$$A = -270a^3, B = -72a^2 c_z, C = 6a(1 + c_z^2).$$

Let $w = z^2$, then

$$Aw^2 + Bw + C = 0.$$

The roots are

$$w_1 = \frac{(9c_z^2 + 5)^{1/2} - 2c}{15a}.$$

and

$$w_2 = \frac{(9c_z^2 + 5)^{1/2} + 2c}{-15a}.$$

If $a > 0$ then w_1 is the positive root, while if $a < 0$ then w_2 is the positive root. Let w be the positive root, then the extreme points are

$$x_1 = w^{1/2} - \frac{b}{3a},$$

and

$$x_2 = -w^{1/2} - \frac{b}{3a}.$$

The curvatures at these points are negatives of each other because of the odd symmetry of the curvature about the inflection point. If a is zero then the maximum curvature occurs at the local extremum of the parabola, which is

$$x = -\frac{c}{2b}.$$

If a and b are zero, the curvature is zero everywhere. See FORTRAN subroutine cubicx and the Maple derivation min15 (4/17/91).

6 Ferguson Cubics

A Ferguson cubic takes the form

$$r(u) = r_0(1 - 3u^2 + 2u^3) + r_1(3u^2 - 2u^3) + \dot{r}_0(u - 2u^2 + u^3) + \dot{r}_1(-u^2 + u^3),$$

where u is a parameter in the interval $[0, 1]$. Then

$$r(0) = r_0, r(1) = r_1.$$

The derivative is

$$\dot{r}(u) = r_0(6u^2 - 6u) + r_1(6u - 6u^2) + \dot{r}_0(1 - 4u + 3u^2) + \dot{r}_1(-2u + 3u^2).$$

Then

$$\dot{r}(0) = \dot{r}_0, \dot{r}(1) = \dot{r}_1.$$

The second derivative is

$$\ddot{r}(u) = r_0(-6 + 12u) + r_1(6 - 12u) + \dot{r}_0(-4 + 6u) + \dot{r}_1(-2 + 6u).$$

7 Hermite Interpolation

Given a set of n points $\{p_i\}$ and n tangent angles $\{\theta_i\}$, we shall construct a c_1 piecewise cubic that interpolates these values. From a tangent angle θ we may construct a unit tangent vector

$$\frac{dr}{ds} = (\cos(\theta), \sin(\theta)).$$

For the Ferguson cubic we need

$$\dot{r} = \frac{dr}{du} = \frac{dr}{ds}\frac{ds}{du}.$$

Consider an interpolation point. Let $\Delta_- s$ and $\Delta_+ s$ be the previous and following chord lengths. Let the average chord length be

$$\Delta s = \frac{\Delta_- s + \Delta_+ s}{2}.$$

The u parameter will vary from 0 to 1 between successive points. Therefore

$$\frac{ds}{du} \approx \frac{\Delta s}{1}.$$

We shall approximate $\dot{r}$ by

$$\frac{dr}{ds}\Delta s.$$

In the case of an endpoint, we shall make the obvious modification. Let the curve parameter t take values on the interval $[0, n-1]$. Then if $k \leq t \leq k+1$, the local parameter is

$$u = t - k.$$

This curve interpolates the given values, and is continuous, and has continuous velocity. Also the chord length approximates the arc length, and the

curve should tend to have the same shape as the data. Note that if the speeds at the knots, the $\|\dot{r}\|$, are arbitrary, then the curve may overshoot and may have cusps or loops, and then the chord length may not approximate the arc length, and the shape may not be the same as the data. We may wish to convert the resulting curve to Bezier or B-spline form.

8 On The Determination Of Maximum Curvature Of A Cubic (Dixon)

Cubic splines have found frequent use in computer aided design and manufactoring. As such it is of interest to be concerned as to the nature of its' curvature, in particular where its' maximum curvature occurs. Towards this end, let us consider a general cubic given by:

$$f(x) = ax^3 + bx^2 + cx + d \tag{1}$$

8.1 The Analysis

The curvature of (1) is defined as:

$$\kappa(x) = \frac{f''(x)}{[1 + f'(x)^2]^{\frac{3}{2}}} \tag{2}$$

We elect not to use the absolute value of the second derivative as we are concerned here about the nature or direction of the curvature as well as its' magnitude.

By setting the derivative of (2) equal to zero, we can determine the critical points of (2), hence locating where its' curvature is maximum. After doing this and some algebra we obtain:

$$f'''(x)[1 + f'(x)^2] - 3f''(x)^2 f'(x) = 0 \tag{3}$$

Substituting the expressions for the first, second and third derivatives of (1) into (3), and more algebra we have a quartic equation given by:

$$a[1 + (3ax^2 + 2bx + c)^2] - 2(3ax + b)^2(3ax^2 + 2bx + c) = 0 \tag{4}$$

Let us consider the function of the left hand member of (4) given by:

$$g(x) = a[1 + (3ax^2 + 2bx + c)^2] - 2(3ax + b)^2(3ax^2 + 2bx + c) \qquad (5)$$

We will prove two properties: first, that $\tilde{x} = \frac{-b}{3a}$ is a root of $g'(x)$ and second, that (4) has a pair of complex roots for which $\tilde{x}$ is the real part. We have:

$$g'(x) = (3ax + b) \cdot G(x) \qquad (6)$$

where,

$$G(x) = -4[a(3ax^2 + 2bx + c) + (3ax + b)^2] \qquad (7)$$

This shows the first property that we wanted to prove, that $g'(\tilde{x}) = 0$. This also implies that $(\tilde{x}, g(\tilde{x}))$ is a critical point of $g(x)$. Closer analysis reveals that $\tilde{x}$ is a root of the second derivative of (1), which means that it is a root of (2). And verification of the second property indicates that it is the value of the independent variable where (1) changes its' concavity.

Let us now write $g(x)$ in standard form as follows:

$$g(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E \qquad (8)$$

where,

$$\begin{aligned} A &= -45a^3 && (9)\\ B &= -60a^2b && (10)\\ C &= -2a(6ac + 13b^2) && (11)\\ D &= -4b(2ac + b^2) && (12)\\ E &= a + ac^2 - 2b^2c && (13) \end{aligned}$$

It is well known that cubics are anti symmetrical about their inflection points. Hence, (1) is antisymmetrical about $\tilde{x}$. We have in (8) a quartic that was derived from setting the derivative of (2) equal to zero and simplifing. Therefore, since 'A' determines whether its' tails will be above or below the x-axis, to prove the second property it suffices to show that $A \cdot G(\tilde{x}) < 0$.

Substituting $\tilde{x} = \frac{-b}{3a}$ into (8) we get,after some algebra:

$g(\tilde{x}) = a[1 + (c - \frac{b^2}{3a})^2]$

Thus, $A \cdot g(\tilde{x}) = -45a^4[1 + (c - \frac{b^2}{3a})^2] < 0$

Q.E.D.

Let the complex roots of (8) be given by:

$$\left.\begin{array}{l} r_1 = \tilde{x} + iv \\ r_2 = \tilde{x} - iv \end{array}\right\} \tag{14}$$

Then one of the quadratic factors of (8) is:

$$q_1(x) = (x - r_1)(x - r_2) = x^2 - 2\tilde{x}x + (\tilde{x}^2 + v^2) \tag{15}$$

Let the other quadratic factor be given by:

$$q_2(x) = \alpha x^2 + \beta x + \gamma \tag{16}$$

Then we have:

$$g(x) = q_1(x) \cdot q_2(x) \tag{17}$$

or

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = [x^2 - 2\tilde{x}x + (\tilde{x}^2 + v^2)][\alpha x^2 + \beta x + \gamma] \tag{18}$$

And after some algebra we get:

$$\begin{aligned} A &= \alpha && (19)\\ B &= \beta - 2\tilde{x}\alpha && (20)\\ C &= \gamma - 2\tilde{x}\beta + \alpha(\tilde{x}^2 + v^2) && (21)\\ D &= \beta(\tilde{x}^2 + v^2) - 2\tilde{x}\alpha && (22)\\ E &= \gamma(\tilde{x}^2 + v^2) && (23) \end{aligned}$$

From (19) and (20) we readily get $\alpha = A$ and β, where

$$\beta = B + 2\tilde{x}A \tag{24}$$

And from (23) we get

$$\tilde{x}^2 + v^2 = \frac{E}{\gamma} \tag{25}$$

Making these substitutions in (21) we obtain:

$$\gamma - 2\tilde{x}(B + 2\tilde{x})A + \frac{AE}{\gamma} = C \tag{26}$$

or

$$\gamma^2 - [2\tilde{x}(B + 2\tilde{x}A)]\gamma + AE = C \tag{27}$$

Using the quadratic formula to solve for γ we get:

$$\gamma = \frac{[2\tilde{x}(B + 2\tilde{x}A) + C] \pm \sqrt{[2\tilde{x}(B + 2\tilde{x}A) + C]^2 - 4AE}}{2} \tag{28}$$

Since α is a non zero function of a, we use (21) to solve for v. Which yields:

$$v = \sqrt{\frac{C - \gamma' + 2\tilde{x}(B + 2\tilde{x}A)}{A} - \tilde{x}^2} \tag{29}$$

Where γ' is the γ obtained from (28) using the sign that causes the v in (29) to be real.

Thus, we have the complex roots of (8) given by (14), and since a factor of (8) is the quadratic expression in (15)

$$x^2 - 2\tilde{x}x + (\tilde{x}^2 + v^2) \tag{30}$$

by dividing (30) into (8) we can obtain the quadratic expression given in (16), which is the other factor of (8). The result of carrying out this operation, or by simply using (19),(20) and (21), yields:

$$Ax^2 + (B + 2A\tilde{x})x + \{[C - A(\tilde{x}^2 + v^2)] + 2\tilde{x}(B + 2A\tilde{x})\} \tag{31}$$

The quadratic formula readily gives the roots of (31), which are the real roots of (8) and the values of the independent variable where the cubic (1) attains maximum curvature. That is $(\hat{x}, \kappa(\hat{x}))$ is a point of maximum curvature of (1), where $\hat{x}$ is a root of (31).

9 The Trapezoid Rule For Numerical Integration

The integral of a real function on an interval $[a, b]$ may be approximated by dividing the interval into n subintervals

$$a = x_1 < x_2 < ... < x_n = b,$$

and taking the sum of the trapezoidal areas formed under the curve at each subinterval. Thus we have approximately

$$\int_a^b f(x)dx = \Delta x \left[\frac{f(a)}{2} + \sum_{i=2}^{n-1} f(x_i) + \frac{f(b)}{2} \right],$$

where Δx is the length of the subintervals. The truncation error of the trapezoid rule is relatively large. When we can evaluate the function freely at any point, the trapezoid rule can be combined with Richardson extrapolation to give an accurate algorithm called Romberg integration.

10 The Truncation Error of the Trapezoid Rule

Consider the integral

$$\int_x^{x+h} f(u)du.$$

Let $g' = f$. Then

$$\begin{aligned} \int_x^{x+h} f(u)du &= g(x+h) - g(x) \\ &= g'(x)h + g''(x)h^2/2! + ... \\ &= f(x)h + f'(x)h^2/2! + ... \end{aligned}$$

and

$$f(x+h) \ = f(x) + f'(x)h + f''(x)h^2 2! + ...$$

Solving for $f'(x)$, we get

$$f'(x) = [f(x+h) - f(x) - f''(x)h^2/2! - ...]/h.$$

Substituting for f', we have

$$\int_x^{x+h} f(u)du = f(x)h+(h/2)[f(x+h)-f(x)-f''(x)h^2/2!-..]+f''(x)h^3/6+...$$

$$= (f(x+h)+f(x))h/2 + f''(x)h^3(1/6-1/4) + ...$$
$$= (f(x+h)+f(x))h/2 - f''(\xi)h^3/12,$$

where

$$x \leq \xi \leq x+h.$$

11 The Adams-Moulton Formula

We will compute the approximate solution to first order differential equation

$$\frac{dx}{dt} = f(t,x),$$

as a discrete sequence

$$\{(t_i, x_i) : i = 1, 2, 3, 4, ...\}.$$

Suppose $x(t)$ is the actual solution of the differential equation. Using the trapezoid rule

$$x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t))dt$$

$$= h(f(t_i, x(t_i)) + f(t_{i+1}, x(t_{i+1}))/2 - (h^3/12)x(\xi).$$

We compute the numerical solution sequence iteratively using the implicit equation

$$x_{i+1} = x_i + (1/2)(f(t_i, x_i) + f(t_{i+1}, x_{i+1}))h.$$

Then

$$x_{i+1} - x(t_{i+1}) = x - x(t_i) + (1/2)(f(t_i, x_i) - f(t_i, x(t_i)))h$$

$$+(1/2)(f(t_{i+1}, x_{i+1}) - f(t_{i+1}, x(t_{i+1})))h$$
$$-(h^3/12)x^{(3)}(\xi).$$

$$= 0 + 0 + \frac{\partial f(t_{i+1}, \eta)}{\partial x}(x_{i+1} - x(t_{i+1})) - (h^3/12)x^{(3)}(\xi).$$

Solving this equation

$$x_{i+1} - x(t_{i+1}) = 1/[1 - \frac{h}{2}\frac{\partial f(t_{i+1}, \eta)}{\partial x}]\frac{h^3}{12}x^{(3)}(\xi).$$

Expanding the first factor as a geometric series we have

$$x_{i+1} - x(t_{i+1}) = \frac{h^3}{12}x^{(3)}(\xi).$$

This is an implicit third order method. The implicit equation for x_{i+1} can be solved by fixed point iteration. The equation takes the form $z = g(z)$ and for small enough h,

$$\left|\frac{dg}{dz}\right| \leq 1.$$

So the iteration converges. This formula, which functions as a corrector, may be used in a predictor-corrector method. The predictor is the formula,

$$x_{i+1} = x_{i-1} + 2hf(t_i, x_i),$$

derived from the central difference approximation to the derivative. Expanding $x(t_{i+1})$ and $x(t_{i-1})$ in a taylor series, and subtracting, we get

$$x(t_{i+1}) - x(t_{i-1}) = 2x'(t_i)h + 2x'''(t_i)h^3/3! + ...$$

Taking $x_{i-1} = x(t_{i-1})$ and $x_i = x(t_i)$ we find

$$x(t_{i+1}) - x_{i+1} = x'''(t_i)h^3/3 + ...$$

So the local truncation error is of order three. The difference of the predictor and corrector gives an estimate of the local truncation error

12 The Euler Method

Consider the differential equation

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

This solution method is the first order approximation

$$x_2 = x_1 + h$$

$$y_2 = y_1 + f(x_1, y_1)h.$$

13 The Taylor Series Method

Consider the differential equation

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

Let us express the solution as a Taylor series

$$y = \sum_{j=0}^{\infty} \frac{y^{(j)}(x_i)h^j}{j!}.$$

We have

$$y''(x) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f$$

Let $f_{i,j}$ be the ijth partial of f. Then we may write

$$y'' = f_1 + f_2 f$$

We may continue this and get

$$y''' = f_{11} + 2f_{12}f + f_{22}f^2 + f_1 f_2 + f_2^2 f,$$

and so on. Thus, knowing $f(x, y)$ we may compute the partial derivatives of f to get the derivatives of y, and then substitute these into the Taylor series to get the next step of y. The truncation error then depends on the the number of terms taken.

14 Runge-Kutta Methods

A Runge-Kutta method gives a solution to the differential equation

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

in the form

$$y_{i+1} = y_i + \sum_{j=1}^{n} a_j k_j$$

where

$$k_1 = hf(x_i, y_i)$$

$$k_2 = hf(x_i + p_1 h, y_i + q_{11} k_1)$$
$$k_3 = hf(x_i + p_2 h, y_i + q_{21} k_1 + q_{22} k_2)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$k_n = hf(x_i + p_{n-1} h, y_i + q_{n-1,1} k_1 + \ldots + q_{n-1,n-1} k_{n-1})$$

We derive these equations by equating terms with the equations given by the Taylor series method.

15 Derivation of a Second Order RungeKutta Method

See Eng 304 notes.

16 The Fourth Order RungeKutta Method

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$k_1 = hf$$
$$k_2 = hf(x_i + h/2, y_i + k_1/2)$$
$$k_3 = hf(x_i + h/2, y_i + k_2/2)$$
$$k_4 = hf(x_i + h, y_1 + k_3)$$

The error per step is of order h^5.

A Fortran subroutine for a single step:

```
c+ rk4je runge-kutta differential equation step
      subroutine rk4je(y,dydx,n,x,h,yout,derivs)
      implicit real*8(a-h,o-z)
c     modified version of rk4 (numerical recipes)
c  input:
c  y      initial values of vector function
c  dydx   initial values of vector derivative
c  n      number of first order equations
```

```
c         y, dydx are n vectors.
c  x      independent variable
c  h      independent variable step
c  output:
c  yout   value of vector function after step h
c  derivs name of external subroutine that
c         calculates vector derivative.
c  Equation that is solved:
c  dy/dx = f(x,y), where f is defined by subroutine derivs
      parameter (nmax=20)
      dimension y(*),dydx(*),yout(*),yt(nmax),dyt(nmax),dym(nmax)
      external derivs
      hh=h*0.5
      h6=h/6.
      xh=x+hh
      do i=1,n
       yt(i)=y(i)+hh*dydx(i)
      enddo
      call derivs(xh,yt,dyt)
      do i=1,n
        yt(i)=y(i)+hh*dyt(i)
      enddo
      call derivs(xh,yt,dym)
      do i=1,n
       yt(i)=y(i)+h*dym(i)
       dym(i)=dyt(i)+dym(i)
      enddo
      call derivs(x+h,yt,dyt)
      do i=1,n
       yout(i)=y(i)+h6*(dydx(i)+dyt(i)+2.*dym(i))
      enddo
      return
      end
```

A C++ program example:

```
//rk4je.cpp
#include <stdio.h>
```

```
#include <stdlib.h>
#include <math.h>
main(){
 void rk4je(double *,double *,int,double,double,
 double *,void (*)(double,double *,double *));
 void der1(double,double *,double *);
 int i,m,n,np,npa,k;
 double y[2],dydx[2],yout[2],x,xend,h;
 //c     starting value of independent variable:
 x=0.;
 #define pi 3.14159265358979
 //c     ending value of independent variable:
 xend=pi/2.;
 //c     approximate step size:
 h=.001;
 //c     number of steps:
 m=(xend-x)/h;
 //c     adjusted step size:
 h=(xend-x)/m;
 //c     approximate number of print points:
 npa=15;
 //c     steps between prints:
 np=m/npa;
 if(np < 1)np=1;
 n=2;
 y[0]=0.;
 y[1]=1.;
 printf("%d %21.14g %21.14g %21.14g\n",0,x,y[0],sin(x));
 for(i=1;i<=m ;i++){
  //c      calculate derivatives of y at x
  der1(x,y,dydx);
  rk4je(y,dydx,n,x,h,yout,der1);
  //c      update values:
  x=x+h;
  y[0]=yout[0];
  y[1]=yout[1];
  k= i % np;
```

```
  if((k  ==  0)  ||  (i  ==  m)){
   printf("%d %21.14g %21.14g %21.14g\n",i,x,y[0],sin(x));
  }
 }
}
//c+ rk4je runge-kutta differential equation step
 void rk4je(double *y,double *dydx,int n,double x,double h,
  double *yout,void (*derivs)(double,double *,double *)){
 //c     modified version of rk4 (numerical recipes)
 //c  input:
 //c  y      initial values of vector function
 //c  dydx   initial values of vector derivative
 //c  n      number of first order equations
 //c         y, dydx are n vectors.
 //c  x      independent variable
 //c  h      independent variable x step
 //c  output:
 //c  yout   value of vector function after step h
 //c  derivs name of external subroutine that
 //c         calculates vector derivative.
 //c  Equation that is solved:
 //c  dy/dx = f(x,y), where f is defined by subroutine derivs
 double hh,h6,xh;
 int i;
 #define nmax 20
 double yt[nmax],dyt[nmax],dym[nmax];
 hh=h*0.5;
 h6=h/6.;
 xh=x+hh;
 for(i=0;i<n;i++){
  yt[i]=y[i]+hh*dydx[i];
 }
 derivs(xh,yt,dyt);
 for(i=0;i<n ;i++){
  yt[i]=y[i]+hh*dyt[i];
 }
 derivs(xh,yt,dym);
```

```
 for(i=0;i<n ;i++){
  yt[i]=y[i]+h*dym[i];
  dym[i]=dyt[i]+dym[i];
 }
 derivs(x+h,yt,dyt);
 for(i=0;i<n ;i++){
  yout[i]= y[i]+h6*(dydx[i]+dyt[i]+2.*dym[i]);
 }
}
//c+ der1 derivative values, right side of differential equation system
void der1(double x,double y[],double dy[]){
 //External Functions:
 //c     Second order differential equation
 //c     d^2 w/dx^2 + w = 0,
 //c     Initial conditions: w(0)=0, dw(0)/dx = 1
 //c     Exact solution: w(x) = sin(x)
 //c     To convert to a first order system of two equations
 //c     let new variables y_i equal the derivatives:
 //c     y_1=w, y_2= dw/dx
 //c     Then we get a system of two first order equations:
 //c     d y_1/dx = f_1(x,y) =  y_2
 //c     d y_2/dx = f_2(x,y) = -y_1
 dy[0]=y[1];
 dy[1]=-y[0];
}
```

17 Richardson Extrapolation

17.1 A Single Extrapolation

Suppose $q > p$ and

$$F(h) = a_0 + a_1 h^p + O(h^q).$$

We want to calculate a_0 as the limit of $F(h)$ as h goes to zero. We have

$$F(2h) = a_0 + a_1 2^p h^p + O(h^q).$$

We may take a linear combination of these two functions to get a function of order q in h. Thus

$$\frac{2^p F(h) - F(2h)}{2^p - 1} = a_0 + O(h^q).$$

Define

$$F_2(h) = \frac{(2^p - 1) + 1)F(h) - F(2h)}{2^p - 1} = F(h) + \frac{F(h) - F(2h)}{2^p - 1}.$$

Then

$$F_2(h) = a_0 + O(h^q).$$

Hence for h small, $F_2(h)$ is an improved approximation to a_0. The technique can be repeated.

17.2 Repeated Richardson Extrapolation

Suppose

$$F(h) = a_0 + a_1 h^{p_1} + a_2 h^{p_2} + a_3 h^{p_3} + ...$$

where

$$p_1 < p_2 < p_3 < ....$$

Define

$$F_1(h) = F(h),$$

and

$$F_{k+1}(h) = F_k(h) + \frac{F_k(h) - F_k(2h)}{2^{p_k} - 1}.$$

Proposition.

$$F_k(h) = a_0 + a_k^{(k)} h^{p_k} + a_{k+1}^{(k)} h^{p_{k+1}} + ..$$

Proof. This is proved by induction. Assuming the result for k, we find that the term involving h^{p_k} in $F_{k+1}(h)$ is

$$a_k^{(k)} h^{p_k} + \frac{a_k^{(k)} h^{p_k} - a_k^{(k)} 2^{p_k} h^{p_k}}{2^{p_k} - 1}$$

$$= a_k^{(k)} h^{p_k} + \frac{a_k^{(k)} h^{p_k}(1 - 2^{p_k})}{2^{p_k} - 1} = 0.$$

Thus

$$a_k^{(k+1)} = 0.$$

This proves the proposition.

18 Common Applications

Two common applications of Richardson extrapolation include numerical differentiation and numerical integration. The central difference approximation for a derivative and the trapezoid approximation to an integral, both with step size h, deliver their values as a limit as h goes to zero, where the nonzero coefficients of the powers of h are even. That is

$$p_1 = 2, p_2 = 4, ..., p_k = 2k.$$

So

$$2^{p_k} = 4^k - 1.$$

The calculations may be displayed in tabular form

$$\begin{array}{cccc} F_1(h) & & & \\ F_1(h/2) & F_2(h/2) & & \\ F_1(h/4) & F_2(h/4) & F_3(h/4) & \\ ... & ... & ... & ... \end{array}$$

The calculation is stopped when sufficient accuracy has been reached as determined by the differences of neighboring values in the table.

19 Solving Differential Equations Using Finite Differences

Consider Laplaces' equation in two dimensions:

$$\nabla^2 f = 0.$$

That is

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0.$$

Expanding f as a Taylor series , we have

$$f(x+h,y) = f(x,y) + \frac{\partial f}{\partial x}h + \frac{1}{2!}\frac{\partial^2 f}{\partial x^2}h^2 + ...$$

$$f(x-h,y) = f(x,y) - \frac{\partial f}{\partial x}h + \frac{1}{2!}\frac{\partial^2 f}{\partial x^2}h^2 + ...$$

$$f(x,y+h) = f(x,y) + \frac{\partial f}{\partial y}h + \frac{1}{2!}\frac{\partial^2 f}{\partial y^2}h^2 + ...$$

$$f(x,y-h) = f(x,y) - \frac{\partial f}{\partial y}h + \frac{1}{2!}\frac{\partial^2 f}{\partial y^2}h^2 + ...$$

If we neglect terms in h of order higher than 2, and add these four equations, we get

$$f(x,y) = \frac{1}{4}(f(x+h,y) + f(x-h,y) + f(x,y+h) + f(x,y-h)).$$

Then if we divide the domain into a grid with increment h, f is equal to the average of its four neighbors. We can solve the problem by computing this average at each grid point, over and over, until we get convergence.

20 Central Force Example

See Boyce and DiPrima p65 for related escape velocity problem. Consider the simple differential equation of a one dimensional central force.

$$\frac{dx^2}{dt^2} = -\frac{1}{x^2}.$$

The velocity is

$$v = \frac{dx}{dt}.$$

Then

$$\frac{dv}{dt} = \frac{dv}{dx}\frac{dx}{dt} = v\frac{dv}{dx}.$$

Then

$$v\frac{dv}{dx} = -\frac{1}{x^2}.$$

$$vdv = -\frac{dx}{x^2}.$$

So

$$\frac{v^2}{2} = \frac{1}{x} + c$$

$$c = \frac{v_0^2}{2} - \frac{1}{x_0}.$$

$$v = \sqrt{\frac{2(1+cx)}{x}}$$

$$\frac{dt}{dx} = \frac{1}{v} = \sqrt{\frac{x}{2(1+cx)}}.$$

$$t = \int_{x_0}^{x} \sqrt{\frac{u}{2(1+cu)}} du.$$

See **moon.tex** for Euler numerical solution to related problem.

21 Stiff Equations

As was shown above, a nonlinear equation may be locally approximated by a linear equation. If the eigenvalues of the Jacobian are distinct, then the Jacobian can be diagonalized, and the equations become uncoupled. When the eigenvalues are not distinct, we can work with the Jordan normal form of the matrix. But for puposes of analysis, if the multiple roots of the characteristic equation are changed to give distinct nearly equal roots, we have a good approximation to the original problem, whose jacobian can be diagonalized. Then we are dealing with a one dimensional linear equation. We will show how Eulers method applies to the linear approximation to a first order differential equation. We have

$$x_{n+1} = x_n + hf(t_n, x_n).$$

From Taylor's theorem,

$$x(t_{n+1}) = x(t_n) + x'(t_n)h + x''(\xi)h^2/2,$$

where

$$t_n \leq \xi t_{n+1}.$$

The truncation error is $x''(\xi)h^2/2$. The step size h is chosen to satisfy

$$\epsilon = \|x''(\xi)h^2/2\|,$$

where ϵ is the tolerance for the truncation error. For a system of equations there will be a set of such relations and the largest value satisfying all of them will become the step size h. As the solution continues, the step size will change. Suppose one of the equations of the system has a large negative eigenvalue, then the variable will quickly decay to zero. Then h should increase to solve the system in a reasonable time. But as we shall show there is also a stability requirement that does not allow the step size to be increased. Consider the following example, which is from chemical kinetics. Consider the reactions,

$$A + B \rightarrow C$$

and

$$C + D \rightarrow A,$$

which lead to the differential equations

$$dB/dt = -k_1 AB$$

and

$$dD/dt = -k_2 CD,$$

where

$$A = A_0 - (B - B_0) + (D - D_0)$$

and

$$C = C_0 + (B - B_0) - (D - D - 0).$$

Note that $A + C$ is constant. Suppose

$$A_0 = 2, C_0 = 5, B_0 = .001$$

and

$$D_0 = .0001.$$

Then

$$dB/dt = -(k_1 A_0)B$$

and

$$dD/dt = -(k_2 C_0)D.$$

The solutions are

$$B = B_0 exp(-k_1 A_0 t)$$

and

$$D_0 = D\ exp(-k_2 C_0 t).$$

Suppose $k_1 = 1000$ and $k_2 = 1$. Using Eulers method,

$$B_{n+1} = (1 - 2000h)B_n$$

and

$$D_{n+1} = (1 - 5h)D_n.$$

$B(t)$ goes rapidly to zero and B_n gets very small. But h can not be increased to a reasonable value suitable for the second equation. Because if $\|1 - 2000h\| > 1$, then B_{n+1} will be magnified, and will eventually become arbitrarily large. This is the stiffness problem. The global error is defined to be

$$\delta = x_n - x(t_n).$$

We have

$$\begin{aligned} x_{n+1} &= x_n + ((x_n - g(t_n))J + g'(t_n))h. \\ &= \delta_n + x(t_n) + ((\delta_n + x(t_n) - g(t_n))J + g'(t_n))h, \\ &= \delta_n 1 + x(t_n) + \delta_n hJ + x'(t_n)h \\ &= \delta_n(1 + Jh) + x(t_n) + x'(t_n)h \\ \delta_{n+1} &= \delta_n(1 + Jh) + (x(t_n) + x'(t_n)h - x(t_{n+1}. \end{aligned}$$

The global error is amplified unless either $-1 \leq 1 + Jh \leq 1$ or, $-2 \leq Jh \leq 0$. This controls stability. Consider the backward Euler method,

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1}).$$

We find

$$\delta_{n+1} = (1 - hJ)^{-1}\delta_n + (1 - hJ)^{-1}h^n x''(t_n)/2 + O(h^3).$$

The error is damped when $\|1/(1 - hJ)\| \leq 1$, which holds for J in the left half plane. So there is no longer a stability limitation. Methods for stiff equations are implicit. Often fixed point iteration can not be used to solve the implicit equation. Some form of Newton's method is used in this case. The Jacobian must be repeatedly calculated. This can be expensive.

22 Eigenvalues

Given a matrix A, let A^* be the complex conjugate of the transpose. A matrix is hermitian if $A = A^*$. A real hermitian matrix is called a symmetric matrix and then $A = A^T$. The inner product of a complex vector X and a complex vector Y is , $(X, Y) = X^*Y$. The norm or length of a vector X is $\|X\| = \sqrt{(X, X)}$,

$$(X, X) = \bar{x}_1 x_1 + ... + \bar{x}_n x_n = |x_1|^2 + ... + |x_n|^2.$$

Proposition. The eigenvalues of a hermitian matrix are real.
Proof.

$$\lambda(X, X) = (X, \lambda X) = (X, AX) = (A^*X, X) = (AX, X) = (\lambda X, X) = \bar{\lambda}(X, X).$$

Hence $\lambda = \bar{\lambda}$ so λ is real.
Proposition. The eigenvectors corresponding to distinct eigenvalues are linearly independent.
Proposition. If A is Hermitian (symmetric) and eigenvalues λ_1 and λ_2 are not equal, then corresponding eigenvalues are orthogonal. That is

$$(V_1, V_2) = 0.$$

The Spectral Theorem For Real Symmetric Matrices. If A is symmetric then there exists a diagonal matrix D of eigenvalues and an orthogonal matrix V whose columns are eigenvectors so that,

$$AV = VD$$

and

$$V^T AV = D.$$

Note that a matrix is orthogonal if $V^T = V^{-1}$ This is equivalent to the column vectors being orthonormal. An orthogonal transformation preserves the length of vectors.

23 The Jacobi Method For Locating Eigenvalues

Given a symmetric matrix A. Let R be a rotation matrix that transforms the element of A in the pth row and qth column to zero. It is defined by

$$r_{pq} = \sin(\phi), r_{qp} = -\sin(\phi), r_{pp} = \cos(\phi), r_{qq} = \cos(\phi)$$

all other elements on the main diagonal are 1, and all remaining elements of R are zero. The angle ϕ is chosen so that the element of A' in the pth row and qth column is zero, where

$$A' = R^T AR.$$

In what follows let

$$s = \sin(\phi)$$

and

$$c = \cos(\phi).$$

Let $B = AR$, then $A' = R^T B$. Then

$$a'_{pq} = \sum r_{ip} b_{iq} = \sum r_{ip} \sum a_{ij} r_{jq}$$

$$= r_{pp}(a_{pp} r_{pq} + a_{pq} r_{qq}) + r_{qp}(a_{qp} r_{pq} + a_{qq} r_{qq})$$

$$= c(a_{pp} s + a_{pq} c) - s(a_{qp} + a_{qq} c)$$

$$= sc(a_{pp} - a_{qq} + a_{pq}(c^2 - s^2) = 0.$$

Rearranging the last equation,

$$(c^2 - s^2)/(2sc) = (a_{pp} - a_{qq})/(2a_{pq}) = \alpha.$$

Then with α defined by this equation and t defined to be s/c, we get a quadratic equation for t.

$$t^2 + 2\alpha t - 1 = 0.$$

If $\alpha = 0$, then we take the root to be 1, otherwise the root of largest magnitude is,

$$-\alpha - \text{sign}(\alpha)\sqrt{(\alpha^2 + 1)} = -\text{sign}(\alpha)(|\alpha| + \sqrt{\alpha^2 + 1}).$$

Dividing the constant coefficient, which is -1, by this root we get the root of smallest magnitude,

$$\frac{\text{sign}(\alpha)}{(|\alpha| + \sqrt{\alpha^2 + 1})}.$$

The magnitude of the root t is thus less than 1. So the rotation angle, of which t is the tangent, is less than $\pi/4$. The tangent is t, so

$$c = 1/(t^2 + 1)$$

and

$$s = ct.$$

We define a sequence of matrices by, $A_0 = A$, and $A_{n+1} = R_n^T A_n R_n$, where R_n is a rotation matrix that zeroes some off-diagonal element of A . The sequence $\{A_n\}$ converges to a diagonal matrix D. Let $V = R_1 R_2 R_3 ....R_n ...$, then,

$$D = V^T AV.$$

V is an orthogonal matrix because it is the product of orthogonal matrices. Hence $V^T = V^{-1}$ and so $AV = VD$, which shows that the diagonal elements of D are the eigenvalues of A with the columns of V being the corresponding eigenvectors.

To show that the sequence does converge, we define S to be the sum of the squares of the off-diagonal elements of A. Evaluating the components of A', we find

$$a'_{rp} = ca_{rp} - sa_{rp}$$

and

$$a'_{rq} = ca_{rq} + sa_{rp}$$

for r not q or not p. Squaring and adding,

$$(a'_{rp})^2 + (a'_{rq})^2 = (c^2 + s^2)(a_{rp}^2 + a_{rq}^2)$$

$$= (a_{rp}^2 + a_{rq}^2)$$

So the sum of the squares of the off-diagonal elements is invariant, except for a_{pq}^2 and a_{qp}^2, which are replaced by zero. Thus $S' = S - 2a_{pq}^2$. So $\{S_n\}$ is a decreasing sequence bounded by zero. So it converges. We may zero the

off-diagonal element of largest magnitude at each step and thus make S_{n+1} decrease by at least a fixed proportion:

$$S_{n+1} \leq \beta S_n \leq \beta^n S_1,$$

where $\beta < 1$. So $\{S_n\}$ converges to zero, which implies that $\{A_n\}$ converges to a diagonal matrix D.

If we carry out this whole calculation in a slightly different way, zeroing the off-diagonal elements in a cyclic order, then we still get convergence to zero. And then we do not need to find the largest element.

Note. The Jacobi method proves the spectral theorem for symmetric matrices.

24 Leverrier's Method for the Characteristic Polynomial

This method uses Newton's identities and the trace of powers of the matrix. Given a polynomial,

$$x^n + c_1 x^{n-1} + ... + c_1 = (x - \alpha_1)(x - \alpha_2)....(x - \alpha_n).$$

Define $s_k = \sum \alpha_i^k$ to be the sum of the kth powers of the roots. The following equations are known as Newton's identities:

$$s_1 + c_1 = 0$$

$$s_2 + c_1 s_1 + 2c_2 = 0$$

$$s_3 + c_1 s_2 + c_2 s_1 + 3c_3 = 0$$

$$.....$$

$$s_n + c_1 s_{n-1} + ... + c_{n-1} s_1 + n c_n = 0$$

For a proof see a classic algebra book such as: **The Theory of Equations**, by J.V. Uspensky. These identities allow the systematic determination of the coefficients of the polynomial from its roots. Definition: The trace of a matrix is the sum of its diagonal elements.
Proposition. Trace(AB) = Trace(BA).

Proof.

$$\text{Trace}(AB) = \sum\sum A_{ik}B_{ki} \sum\sum B_{ki}A_{ik} = \text{Trace}(BA).$$

From this proposition it follows that, + -1

$$\text{Trace}(B^{-1}AB) = \text{Trace}(A).$$

Proposition. Let matrix A have eigenvalues $\lambda_1, ..., \lambda_n$. Then

$$\text{Trace}(A^k) =\sim \lambda_i^k = s_k.$$

The steps of Leverrier's method are: (1) Compute A^k and $s_k = \text{Trace}(A^k)$, (2) Use Newton's identities repeatedly to get the coefficients of the characteristic polynomial.

Faddeev's modification of the method consists in a more efficent calculation method. Refer to the book by James, Smith, and Wohlford.

25 Bibliography

[1] Dahlquist Germund, Björck Ake, **Numerical Methods**, 1974, Prentice-Hall.

[2] James M L, Smith G M, Wolford J C, **Applied Numerical Methods for Digital Computers** Third Edition, 1985, Harper and Row.

[3]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

[1]

26 Appendix A, Numerical Analysis Libraries and Software