Numerical Analysis

Dr JSC Prentice

October 15, 2017

Contents

1 Newton's Method

1.1 Fixed Points

The number p is a *fixed point* of $g(x)$ if

$$g(p) = p.$$

If $g \in C[a,b]$ and $g(x) \in [a,b]$, then g has at least one fixed point in $[a,b]$. If, in addition, $g'(x)$ exists on (a,b) and

$$|g'(x)| < 1$$

for all $x \in (a,b)$, then the fixed point is unique.

1.2 Functional Iteration

Using some initial value p_0, we can define a sequence $\{p_n\}_{n=0}^{\infty}$ by

$$p_n = g(p_{n-1})$$

for $n \geqslant 1$. This process is known as *functional iteration.*

1.3 The Fixed Point Theorem

Assume that $g(x)$ has a unique fixed point p in $[a,b]$. Then, for any $p_0 \in [a,b]$, functional iteration of $g(x)$ will converge to p. In other words,

$$\lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = p.$$

1.4 Convergence

Define

$$e_n \equiv p_n - p,$$

so that $p_{n-1} = e_{n-1} + p$. Now, we have

$$\begin{aligned} e_n &= g(p_{n-1}) - p \\ &= g(e_{n-1} + p) - p \\ &= g(p) + e_{n-1}g'(p) + \frac{e_{n-1}^2}{2}g''(p) + \frac{e_{n-1}^3}{6}g'''(p) + \ldots - p \\ &= e_{n-1}g'(p) + \frac{e_{n-1}^2}{2}g''(p) + \frac{e_{n-1}^3}{6}g'''(p) + \ldots \end{aligned}$$

Note that, if $g'(p) = 0$, we have

$$e_n = \frac{e_{n-1}^2}{2} g''(p) + \ldots, \tag{1}$$

and if $g'(p) = g''(p) = 0$, we have

$$e_n = \frac{e_{n-1}^3}{6} g'''(p) + \ldots. \tag{2}$$

Suppose that $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges to p, but $p_n \neq p$ for all n. If positive constants α and λ exist with

$$\lim_{n \to \infty} \frac{|e_{n+1}|}{|e_n|^{\alpha}} = \lambda,$$

then we say that $\{p_n\}_{n=0}^{\infty}$ converges to p of *order* α, with *asymptotic error constant* λ. When $g'(p) \neq 0$, we have $\alpha = 1$ and we say that convergence is *linear*. For the case in (1), $\alpha = 2$ (*quadratic* convergence) and $\lambda = \frac{|g''(p)|}{2}$. For the case in (2), $\alpha = 3$ (*cubic* convergence) and $\lambda = \frac{|g'''(p)|}{6}$.

1.5 Construction of Newton's Method

We can exploit the above ideas to construct Newton's Method for finding the root of a function $f(x) \in C^{\infty}$. We seek to use functional iteration with quadratic convergence. A suitable definition of the function $g(x)$ is

$$g(x) = x - \phi(x) f(x),$$

where $\phi(x)$ is a function to be determined. Note that, if p is the root of $f(x)$,

$$g(p) = p - \phi(p) f(p) = p,$$

so that p is a fixed point of $g(x)$. Functional iteration of $g(x)$, if it converges, will converge to p.

To find $\phi(x)$, we demand that functional iteration of $g(x)$ converges *quadratically*. From the above, this requires

$$g'(p) = 0$$

which gives

$$\begin{aligned} g'(p) &= 1 - \phi'(p) f(p) - \phi(p) f'(p) \\ &= 1 - \phi(p) f'(p) \\ &= 0 \\ \Rightarrow \phi(p) &= \frac{1}{f'(p)}. \end{aligned}$$

This implies that

$$\phi(x) = \frac{1}{f'(x)}$$

is the appropriate form for $\phi(x)$. We now have

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

or, in the notation of functional iteration,

$$p_n = g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \tag{3}$$

which is Newton's Method.

A condition for convergence can be found by demanding that

$$|g'(x)| < 1.$$

This gives

$$-1 < \frac{f(x) f''(x)}{[f'(x)]^2} < 1.$$

Note that this inequality is satisfied when $x = p$. Hence, any neighbourhood of p on which this inequality is satisfied can be taken as the interval $[a, b]$ in the Fixed Point Theorem. If p_0 is then chosen within this interval, convergence to p is guaranteed. Note also that, since $f(x)$ and its derivatives are continuous (by assumption), such a neighbourhood of p does exist.

1.6 Root Multiplicity

A root p of $f(x)$ is said to have *multiplicity* m if

$$f(x) = (x - p)^m q(x),$$

where $\lim_{x \to p} q(x) \neq 0$. It can be shown that $f(x) \in C^m[a, b]$ has a zero of multiplicity m at p in $[a, b]$ iff

$$f(p) = f'(p) = \ldots = f^{(m-1)}(p) = 0$$

and

$$f^{(m)}(p) \neq 0.$$

Newton's Method does not converge quadratically to roots of multiplicity $m > 1$. In order to recover quadratic convergence, the method must be modified. Two such modifications are

$$p_n = p_{n-1} - \frac{f^{(m-1)}\left(p_{n-1}\right)}{f^{(m)}\left(p_{n-1}\right)}$$

$$p_n = p_{n-1} - \frac{mf\left(p_{n-1}\right)}{f'\left(p_{n-1}\right)}.$$

2 Exercises

1. Find the fixed points of $g(x) = 2x - Ax^2$, where $A > 0$. Then find an interval about one of these fixed points, such that functional iteration of $g(x)$ will converge to this fixed point.

2. Show that if A is any positive number, then the sequence

$$x_n = \frac{x_{n-1}}{2} + \frac{A}{2x_{n-1}}$$

converges to $\sqrt{A}$ whenever $x_0 > 0$. What happens when $x_0 < 0$?

3. Show that the method

$$x_n = x_{n-1} - \frac{f^{(m-1)}(x_{n-1})}{f^{(m)}(x_{n-1})}.$$

converges quadratically to a root of multiplicity m.

4. Show that the method

$$x_n = x_{n-1} - \frac{mf(x_{n-1})}{f'(x_{n-1})}.$$

converges quadratically to a root of multiplicity m.

5. Derive the method

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} - \frac{f''(x_{n-1})[f(x_{n-1})]^2}{2[f'(x_{n-1})]^3}$$

by demanding cubic convergence for

$$x_n = g(x_{n-1}),$$

where $g(x)$ has a suitable form.

6. Confirm that the method

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} - \frac{f''(x_{n-1})[f(x_{n-1})]^2}{2[f'(x_{n-1})]^3}$$

is cubically convergent.

3 Solutions

1. We have

$$g(x) = x \;\Rightarrow\; 2x - Ax^2 = x \;\Rightarrow\; x = Ax^2 \;\Rightarrow\; x = \frac{1}{A}.$$

So the fixed point is $p = \frac{1}{A}$. We seek an interval D such that $|g'(x)| < 1$ on D and $g(D) \subseteq D$ and $\frac{1}{A} \in D$. Now, $g'(x) = 2 - 2Ax$ so that

$$|2 - 2Ax| < 1 \;\Rightarrow\; \frac{1}{2A} < x < \frac{3}{2A}.$$

We are assuming $A > 0$. Now, $g(x)$ is continuous and we have

$$g\left(\frac{1}{2A}\right) = \frac{3}{4A}$$
$$g\left(\frac{3}{2A}\right) = \frac{3}{4A}$$
$$g\left(\frac{1}{A}\right) = \frac{1}{A}$$

and $g\left(\frac{1}{A}\right)$ is a maximum. So

$$\frac{3}{4A} < \frac{1}{A}$$

and

$$g\left(\left(\frac{1}{2A}, \frac{3}{2A}\right)\right) = \left(\frac{3}{4A}, \frac{1}{A}\right).$$

Moreover,

$$\left(\frac{3}{4A}, \frac{1}{A}\right) \subseteq \left(\frac{1}{2A}, \frac{3}{2A}\right)$$

and so $g(x)$ maps the interval $D \equiv \left(\frac{1}{2A}, \frac{3}{2A}\right)$ into itself. Certainly, $\frac{1}{A}$ is contained within D and $|g'(x)| < 1$ on D, so by the Fixed-Point theorem convergence to $\frac{1}{A}$ is guaranteed, provided that the initial point is in D. Furthermore, the fixed point $\frac{1}{A}$ is unique.

2. Firstly, $g\left(\sqrt{A}\right) = \sqrt{A}$ so that $\sqrt{A}$ is a fixed point of g. Secondly, $g'(x) = 1/2 - A/2x^2$ so that $|g'(x)| < 1$ for all $x > \sqrt{A}$. Note that $g'(x) = 0 \Rightarrow x = \sqrt{A}$ so that g has its only minimum on $\left[\sqrt{A}, \infty\right)$ at $\sqrt{A}$. Hence, g is strictly increasing on $\left[\sqrt{A}, \infty\right)$ in general, and on

$\left[\sqrt{A}, x_0\right]$ in particular. Now, $g(x_0) = \frac{x_0}{2} + \frac{A}{2x_0} < x_0$ when $x_0 > \sqrt{A}$. Consider $z \in \left[\sqrt{A}, x_0\right]$. Then $\sqrt{A} \leqslant g(z) \leqslant g(x_0)$ because g is strictly increasing, and, since $g(x_0) < x_0$ we have $\sqrt{A} \leqslant g(z) \leqslant g(x_0) < x_0$. This means that $g\left(\left[\sqrt{A}, x_0\right]\right) \subseteq \left[\sqrt{A}, x_0\right]$. As a result, the hypotheses of the Fixed-Point Theorem are satisfied on $\left[\sqrt{A}, x_0\right]$, and so functional iteration $x_m = g(x_{m-1})$, with $x_0 > \sqrt{A}$, will converge to the fixed point $x = \sqrt{A}$. If $0 < x_0 < \sqrt{A}$ we have

$$\begin{aligned} & 0 < \left(x_0 - \sqrt{A}\right)^2 = x_0^2 - 2x_0\sqrt{A} + A \\ \Rightarrow\ & 2x_0\sqrt{A} < x_0^2 + A \\ \Rightarrow\ & \sqrt{A} < \frac{x_0}{2} + \frac{A}{2x_0} = g(x_0) = x_1. \end{aligned}$$

Hence: $0 < x_0 < \sqrt{A} \Rightarrow \sqrt{A} < x_1$, by part (3). Hence,

$$0 < x_0 < \sqrt{A} < x_{m+1} < x_m < \cdots < x_1$$

and so $\lim_{m\to\infty} x_m = \sqrt{A}$. Also, $x_0 = \sqrt{A} \Rightarrow x_m = \sqrt{A}$ for all m. Lastly, $x_0 > \sqrt{A} \Rightarrow \lim_{m\to\infty} x_m = \sqrt{A}$, by part (1). If $x_0 < 0$ we have $\lim_{m\to\infty} x_m = -\sqrt{A}$.

3. Define

$$F(x) \equiv x - \frac{f^{(m-1)}(x)}{f^{(m)}(x)}$$

Note that $F(p) = p$. Hence,

$$x_{n+1} - p = F(x_n) - F(p).$$

Defining

$$e_n \equiv x_n - p$$

gives

$$\begin{aligned} e_{n+1} &= F(e_n + p) - F(p) \\ &= F(p) + e_n F'(p) + e_n^2 \frac{F''(p)}{2} + e_n^3 \frac{F'''(p)}{6} + \ldots - F(p) \\ &= e_n F'(p) + e_n^2 \frac{F''(p)}{2} + e_n^3 \frac{F'''(p)}{6} + \ldots \end{aligned}$$

It is easy to show that

$$F'(p) = \frac{f^{(m+1)}(p) f^{(m-1)}(p)}{f^{(m)}(p) f^{(m)}(p)} = 0$$
$$F''(p) = \frac{f^{(m+1)}(p)}{f^{(m)}(p)} \neq 0 \text{ in general}$$

and so

$$e_{n+1} = e_n^2 \frac{F''(p)}{2} + \dots$$

indicating quadratic convergence.

4. We have

$$g(x) = x - \frac{m f(x)}{f'(x)}.$$

Assume

$$f(x) = (x-p)^m q(x)$$

where $q(p) \neq 0$. Hence,

$$f'(x) = m(x-p)^{m-1} q(x) + (x-p)^m q'(x).$$

Now, we have

$$g(x) = x - m(x-p)^m q(x) \left[m(x-p)^{m-1} q(x) + (x-p)^m q'(x) \right]^{-1}$$

so that

$$g'(x) = 1 - A - B + CD$$

where

$$A = \frac{m^2 (x-p)^{m-1} q(x)}{m(x-p)^{m-1} q(x) + (x-p)^m q'(x)}$$
$$= \frac{m^2 q}{mq + (x-p) q'}$$

$$B = \frac{m(x-p)^m q'(x)}{m(x-p)^{m-1} q(x) + (x-p)^m q'(x)}$$
$$= \frac{m(x-p) q'}{mq + (x-p) q'}$$

$$\begin{aligned} C &= \frac{m(x-p)^m q(x)}{\left[m(x-p)^{m-1} q(x) + (x-p)^m q'(x)\right]^2} \\ &= \frac{m(x-p)^m q(x)}{(x-p)^{2m-2}\left[mq(x) + (x-p) q'(x)\right]^2} \\ &= \frac{m(x-p)^{2-m} q}{\left[mq + (x-p) q'\right]^2} \end{aligned}$$

$$\begin{aligned} D &= m(m-1)(x-p)^{m-2} q + (x-p)^m q'' + 2m(x-p)^{m-1} q' \\ &= (x-p)^{m-2}\left[m(m-1) q + (x-p)^2 q'' + 2m(x-p) q'\right] \end{aligned}$$

and

$$CD = \frac{mq\left[m(m-1) q + (x-p)^2 q'' + 2m(x-p) q'\right]}{\left[mq + (x-p) q'\right]^2}$$

Note that

$$\begin{aligned} A(p) &= \frac{m^2 q(p)}{mq(p)} = m \\ B(p) &= 0 \\ C(p) D(p) &= \frac{mq(p)\left[m(m-1) q(p)\right]}{\left[mq(p)\right]^2} = m - 1. \end{aligned}$$

Thus,

$$\begin{aligned} g'(p) &= 1 - m - 0 + (m-1) \\ &= 0. \end{aligned}$$

5. Assume

$$g = x - \phi f - \theta f^2.$$

Then

$$g' = 1 - \phi' f - \phi f' - \theta' f^2 - 2\theta f f'$$

and

$$\begin{aligned} g'' = &- \phi' f' - \phi'' f - \phi' f' - \phi f'' - \theta'' f^2 - 2\theta' f f' \\ &- 2\theta' f f' - 2\theta f' f' - 2\theta f f''. \end{aligned}$$

At $x = p$ we have, since $f(p) = 0$,

$$\begin{aligned} 0 = g'(p) &= 1 - \phi(p) f'(p) \\ \Rightarrow \phi(p) = \frac{1}{f'(p)} &\Rightarrow \phi = \frac{1}{f'} \Rightarrow \phi' = -\frac{f''}{\left[f'\right]^2} \end{aligned}$$

and

$$g''(p) = 0 \;\Rightarrow\; \theta = \frac{1}{2[f']^2}\frac{f''}{f'}.$$

Hence,

$$g(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)[f(x)]^2}{2[f'(x)]^3}.$$

6. We have

$$p_{n+1} - p = g'(p)(p_n - p) + \frac{g''(p)}{2}(p_n - p)^2 + \frac{g'''(p)}{6}(p_n - p)^3 + \dots$$

We must show that $g'(p) = 0$ and $g''(p) = 0$ for convergence to be cubic. We find

$$g' = -\frac{1}{2}\frac{f^2 f'''}{(f')^3} + \frac{3}{2}\frac{f^2 (f'')^2}{(f')^4}$$

$$g'' = -\frac{1}{2}\left[\frac{f^{(4)} f^2}{(f')^3} + \frac{2ff'''}{(f')^2} - \frac{3f^2 f'' f'''}{(f')^4}\right] + \frac{3}{2}\left[\frac{2f^2 f'' f'''}{(f')^4} + \frac{2f(f'')^2}{(f')^3} - \frac{4f^2 (f'')^3}{(f')^5}\right]$$

which gives $g'(p) = 0$ and $g''(p) = 0$, since $f(p) = 0$.

4 Newton's Method for Systems

4.1 Notation and Terminology

We define the *vector function* $\mathbf{G}(\mathbf{x})$ as

$$\mathbf{G}(\mathbf{x}) \equiv \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_N(\mathbf{x}) \end{bmatrix},$$

where

$$\mathbf{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}.$$

Clearly, $\mathbf{G}(\mathbf{x})$ is a map from $\mathbb{R}^N$ to $\mathbb{R}^N$. Each $g_i(\mathbf{x})$ is a scalar function of the components of $\mathbf{x}$.

We say that $\mathbf{p} \in \mathbb{R}^N$ is a fixed point of $\mathbf{G}(\mathbf{x})$ if

$$\mathbf{G}(\mathbf{p}) = \mathbf{p}.$$

Now, let D be a bounded, compact region of $\mathbb{R}^N$. Let $\mathbf{G}(\mathbf{x})$ be a continuous map from D to $\mathbb{R}^N$, such that $\mathbf{G}(\mathbf{x}) \in D$ if $\mathbf{x} \in D$. Then $\mathbf{G}(\mathbf{x})$ has a fixed point in D. Moreover, assume that each component of $\mathbf{G}(\mathbf{x})$ has continuous partial derivatives, and that

$$\left|\frac{\partial g_i(\mathbf{x})}{\partial x_j}\right| < \frac{1}{N} \qquad i, j = 1, 2, \ldots, N$$

for all $\mathbf{x} \in D$. Then the sequence

$$\mathbf{x}^{(n)} = \mathbf{G}\left(\mathbf{x}^{(n-1)}\right), \tag{4}$$

with $\mathbf{x}^{(0)} \in D$, converges to the unique fixed point $\mathbf{p} \in D$.

4.2 N-dimensional Newton's Method

Let $\mathbf{p}$ denote the root of the N-dimensional system

$$\mathbf{F}(\mathbf{x}) \equiv \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{bmatrix} = 0.$$

To construct Newton's Method for this system, we find a function $\mathbf{G}(\mathbf{x})$ such that $\mathbf{p}$ is a fixed point of $\mathbf{G}(\mathbf{x})$. Such a function is given by

$$\mathbf{G}(\mathbf{x}) \equiv \mathbf{x} - \Phi(\mathbf{x})\,\mathbf{F}(\mathbf{x}), \tag{5}$$

where $\Phi(\mathbf{x})$ is an $N \times N$ matrix to be determined. Note that

$$\mathbf{G}(\mathbf{p}) \equiv \mathbf{p} - \Phi(\mathbf{p})\,\mathbf{F}(\mathbf{p}) = \mathbf{p}$$

since $\mathbf{F}(\mathbf{p}) = \mathbf{0}$, so that $\mathbf{p}$ is a fixed point of $\mathbf{G}(\mathbf{x})$.

Consider the kth element of (5)

$$g_k(\mathbf{x}) = x_k - \sum_{i=1}^{N} \phi_{ki}(\mathbf{x})\, f_i(\mathbf{x}).$$

The second term on the RHS is the kth row of Φ multiplied by $\mathbf{F}$. The derivative with respect to x_j gives

$$\frac{\partial g_k(\mathbf{x})}{\partial x_j} = \frac{\partial x_k}{\partial x_j} - \sum_{i=1}^{N} \frac{\partial \phi_{ki}}{\partial x_j}(\mathbf{x})\, f_i(\mathbf{x}) - \sum_{i=1}^{N} \phi_{ki}(\mathbf{x})\, \frac{\partial f_i(\mathbf{x})}{\partial x_j}.$$

Imposing the condition

$$\frac{\partial g_k(\mathbf{p})}{\partial x_j} = 0,$$

which is the appropriate condition for *quadratic convergence* of the functional iteration in (4), gives

$$\begin{aligned} 0 &= \frac{\partial x_k}{\partial x_j} - \sum_{i=1}^{N} \frac{\partial \phi_{ki}}{\partial x_j}(\mathbf{p})\, f_i(\mathbf{p}) - \sum_{i=1}^{N} \phi_{ki}(\mathbf{p})\, \frac{\partial f_i(\mathbf{p})}{\partial x_j} \\ &= \delta_{kj} - \sum_{i=1}^{N} \phi_{ki}(\mathbf{p})\, \frac{\partial f_i(\mathbf{p})}{\partial x_j}, \end{aligned}$$

since $f_i(\mathbf{p}) = 0$, and where δ_{kj} is the Kronecker delta. Note that, since $k, j = 1, 2, \ldots, N$, the Kronecker delta is the (k, j) element of the $N \times N$ identity matrix I_N, and the second term on the RHS is the product of the kth row of Φ and the jth column of the Jacobian J of $\mathbf{F}$. This yields the matrix equation

$$0 = I_n - \Phi(\mathbf{p})\, J(\mathbf{p}),$$

which gives

$$\Phi(\mathbf{p}) = J^{-1}(\mathbf{p}).$$

This, in turn, suggests the method

$$\mathbf{x}^{(n)} = \mathbf{G}\left(\mathbf{x}^{(n-1)}\right) = \mathbf{x}^{(n-1)} - J^{-1}\left(\mathbf{x}^{(n-1)}\right)\mathbf{F}\left(\mathbf{x}^{(n-1)}\right), \tag{6}$$

which is the general form of Newton's Method for N-dimensional systems. Setting $N = 1$ yields (3), as expected. Of course, it is implied here that J is invertible.

5 Exercises

1. Construct the N-dimensional Newton's Method using the total differential of the system.

2. Derive Newton's Method for $N = 1$ and $N = 2$ from (6).

6 Solutions

1. We seek the root of the system

$$\mathbf{F}(\mathbf{x}) \equiv \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{bmatrix} = 0.$$

The total differential is

$$df_i = \frac{\partial f_i}{\partial x_1} dx_1 + \frac{\partial f_i}{\partial x_2} dx_2 + ... + \frac{\partial f_i}{\partial x_N} dx_N$$

which gives

$$\begin{bmatrix} df_1 \\ \vdots \\ df_N \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \begin{bmatrix} dx_1 \\ \vdots \\ dx_N \end{bmatrix} = J \begin{bmatrix} dx_1 \\ \vdots \\ dx_N \end{bmatrix},$$

where J is the Jacobian of $\mathbf{F}$. We assume

$$dx_i \approx x_i^{(n+1)} - x_i^{(n)},$$

where $x_i^{(n)}$ is the value of x_i after the nth iteration, and

$$df_i \approx f_i\left(x_1^{(n+1)}, x_2^{(n+1)}, ..., x_N^{(n+1)}\right) - f_i\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right).$$

Assuming that $\left|f_i\left(x_1^{(n+1)}, x_2^{(n+1)}, ..., x_N^{(n+1)}\right)\right| \ll \left|f_i\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right)\right|$ (i.e., that $\left(x_1^{(n+1)}, x_2^{(n+1)}, ..., x_N^{(n+1)}\right)$ is close to the root) gives

$$df_i \approx -f_i\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right)$$

and, hence,

$$\begin{bmatrix} x_1^{(n+1)} \\ \vdots \\ x_N^{(n+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(n)} \\ \vdots \\ x_N^{(n)} \end{bmatrix} - J^{-1} \begin{bmatrix} f_1\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right) \\ \vdots \\ f_N\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right) \end{bmatrix},$$

where J is evaluated at $\left(x_1^{(n)}, x_2^{(n)}, ..., x_N^{(n)}\right)$.

2. With $N = 1$ we have, from (6),

$$\begin{aligned} x_1^{(n)} &= \mathbf{G}\left(x_1^{(n-1)}\right) \\ &= g_1\left(x_1^{(n-1)}\right) \\ &= x_1^{(n-1)} - J^{-1}\left(x_1^{(n-1)}\right)\mathbf{F}\left(x_1^{(n-1)}\right) \\ &= x_1^{(n-1)} - \left(\frac{\partial f_1}{\partial x_1}\right)^{-1}_{x_1^{(n-1)}} f_1\left(x_1^{(n-1)}\right) \\ &= x_1^{(n-1)} - \frac{f_1\left(x_1^{(n-1)}\right)}{f_1'\left(x_1^{(n-1)}\right)}. \end{aligned}$$

With $N = 2$, we have

$$\begin{aligned} \begin{bmatrix} x_1^{(n)} \\ x_2^{(n)} \end{bmatrix} &= \mathbf{G}\left(x_1^{(n-1)}\right) \\ &= \begin{bmatrix} g_1\left(x_1^{(n-1)}, x_2^{(n-1)}\right) \\ g_2\left(x_1^{(n-1)}, x_2^{(n-1)}\right) \end{bmatrix} \\ &= \begin{bmatrix} x_1^{(n-1)} \\ x_2^{(n-1)} \end{bmatrix} - J^{-1}\left(x_1^{(n-1)}, x_2^{(n-1)}\right)\mathbf{F}\left(x_1^{(n-1)}, x_2^{(n-1)}\right) \\ &= \begin{bmatrix} x_1^{(n-1)} \\ x_2^{(n-1)} \end{bmatrix} - \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}^{-1}_{\left(x_1^{(n-1)}, x_2^{(n-1)}\right)} \begin{bmatrix} f_1\left(x_1^{(n-1)}, x_2^{(n-1)}\right) \\ f_2\left(x_1^{(n-1)}, x_2^{(n-1)}\right) \end{bmatrix}. \end{aligned}$$

7 Polynomial Approximation

7.1 Lagrange Interpolation

The *Lagrange interpolating polynomial* $P_n(x)$ of degree n, at most, that interpolates the data $\{f(x_0), f(x_1), \ldots, f(x_n)\}$ at the nodes $\{x_0, x_1, \ldots, x_n\}$, where $x_0 < x_1 < \cdots < x_n$, is given by

$$P_n(x) = \sum_{k=0}^{n} L_{n,k}(x) f(x_k).$$

Here, the $L_{n,k}(x)$ are known as the *Lagrange coefficient polynomials* and are given by

$$L_{n,k}(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)}.$$

These polynomials are of degree n and have the property

$$L_{n,k}(x_j) = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}.$$

The *pointwise error* in Lagrange interpolation is

$$\Delta(x; P_n) \equiv f(x) - P_n(x) = \frac{f^{(n+1)}(\vartheta(x))}{(n+1)!} \prod_{k=0}^{n} (x - x_k),$$

where $x_0 < \vartheta(x) < x_n$. Clearly, we are assuming that $f(x)$ is $n+1$-times differentiable.

Provided that the nodes $\{x_0, x_1, \ldots, x_n\}$ are distinct, $P_n(x)$ exists and is unique.

7.2 Hermite Interpolation

The *Hermite interpolating polynomial* $H_{2n+1}(x)$ of degree $2n+1$, at most, that interpolates the data $\{f(x_0), f(x_1), \ldots, f(x_n)\}$ and $\{f'(x_0), f'(x_1), \ldots, f'(x_n)\}$ at the nodes $\{x_0, x_1, \ldots, x_n\}$, where $x_0 < x_1 < \cdots < x_n$, is given by

$$H_{2n+1}(x) = \sum_{k=0}^{n} H_{n,k}(x) f(x_k) + \sum_{k=0}^{n} \widehat{H}_{n,k}(x) f'(x_k),$$

where

$$H_{n,k}(x) \equiv \left[1 - 2(x - x_k) L'_{n,k}(x_k)\right] L^2_{n,k}(x)$$
$$\widehat{H}_{n,k}(x) = (x - x_k) L^2_{n,k}(x).$$

Both of these polynomials are of degree $2n+1$ and, clearly, are defined in terms of the Lagrange coefficient polynomials.

The *pointwise error* in Hermite interpolation is

$$\Delta\left(x; H_{2n+1}\right) \equiv f\left(x\right) - H_{2n+1}\left(x\right) = \frac{f^{(2n+2)}\left(\xi\left(x\right)\right)}{(2n+2)!} \prod_{k=0}^{n} \left(x - x_k\right)^2,$$

where $x_0 < \xi\left(x\right) < x_n$. We are assuming that $f\left(x\right)$ is $2n+2$-times differentiable.

Like the Lagrange interpolating polynomial, if the nodes $\{x_0, x_1, \ldots, x_n\}$ are distinct, $H_{2n+1}\left(x\right)$ exists and is unique.

7.3 Pointwise Error for Equispaced Nodes

Assume that the nodes $\{x_0, x_1, \ldots, x_n\}$ are equispaced with stepsize h. We can write

$$\begin{aligned} x_k &= x_0 + kh \\ x &= x_0 + sh, \end{aligned}$$

where $k = 0, 1, \ldots, n$ and $s \in [0, n]$. These give

$$\begin{aligned} \Delta\left(x; P_n\right) &= \frac{f^{(n+1)}\left(\vartheta\left(x\right)\right) h^{n+1}}{(n+1)!} \prod_{k=0}^{n} \left(s - k\right) \\ \Delta\left(x; H_{2n+1}\right) &= \frac{f^{(2n+2)}\left(\xi\left(x\right)\right) h^{2n+2}}{(2n+2)!} \prod_{k=0}^{n} \left(s - k\right)^2 \end{aligned}$$

In the light of these error expressions, we could describe Lagrange interpolation as being of order $n+1$, and Hermite interpolation as being of order $2n+2$.

7.4 Error Control via Piecewise Interpolation

Say we wish to approximate $f\left(x\right)$ on an interval $[a, b]$ using either Lagrange or Hermite interpolation, and we are free to choose the number of nodes (this amounts to choosing n) such that the interpolation error is less than or equal to a prescribed tolerance ε. The obvious condition

$$\frac{f^{(n+1)}\left(\vartheta\left(x\right)\right)}{(n+1)!} \prod_{k=0}^{n} \left(x - x_k\right) \leqslant \varepsilon$$

or

$$\frac{f^{(2n+2)}\left(\xi\left(x\right)\right)}{(2n+2)!}\prod_{k=0}^{n}\left(x-x_k\right)^2 \leqslant \varepsilon$$

is of no use because it does not yield an algebraic expression that enables the calculation of an appropriate n.

Rather, we use *piecewise* interpolation to achieve the desired error control. In such an approach, we subdivide $[a,b]$ into N subintervals and then perform interpolation, *with fixed* n, on each subinterval. As will be shown, with a fixed value for n, it is possible to find an algebraic expression for N such that the magnitude of the pointwise error is bounded by ε.

7.4.1 Lagrange

Assume n is fixed and that $[a,b]$ has been subdivided into N subintervals. If we are to find a $P_n\left(x\right)$ on each subinterval, we would need to define $n+1$ nodes on each subinterval. Let us assume that these nodes are equispaced. Then, on each subinterval, two of the nodes are the endpoints of the subinterval and the remaining $n-1$ nodes are equispaced within the subinterval. If the stepsize is h, then each subinterval has a length nh, which gives $b-a=Nnh$. Now, define M_L and S_L by

$$M_L \equiv \max_{[a,b]}\left|f^{(n+1)}\left(x\right)\right|$$

$$S_L \equiv \max_{[0,n]}\left|\prod_{k=0}^{n}\left(s-k\right)\right|.$$

These give

$$\left|\Delta\left(x;P_n\right)\right| \leqslant \frac{M_L S_L h^{n+1}}{(n+1)!}.$$

Demanding that

$$\frac{M_L S_L h^{n+1}}{(n+1)!} \leqslant \varepsilon$$

gives

$$h \leqslant \left(\frac{(n+1)!\varepsilon}{M_L S_L}\right)^{\frac{1}{n+1}}$$

or

$$N \geqslant \left(\frac{b-a}{n}\right)\left(\frac{M_L S_L}{(n+1)!\varepsilon}\right)^{\frac{1}{n+1}}.$$

We choose the smallest integer value of N consistent with this inequality, as in

$$N = \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_L S_L}{(n+1)!\varepsilon} \right)^{\frac{1}{n+1}} \right\rceil,$$

from which we determine

$$h = \frac{b-a}{Nn}.$$

7.4.2 Hermite

A similar approach holds for Hermite interpolation, where we seek an $H_{2n+1}(x)$ on each subinterval. Defining

$$M_H \equiv \max_{[a,b]} \left| f^{(2n+2)}(x) \right|$$

$$S_H \equiv \max_{[0,n]} \left| \prod_{k=0}^{n} (s-k)^2 \right|$$

gives

$$N = \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_H S_H}{(2n+2)!\varepsilon} \right)^{\frac{1}{2n+2}} \right\rceil.$$

7.4.3 Absolute versus Relative Error

The error control described above holds for the *absolute error* $f(x) - P_n(x)$. If we wish to control the *relative error in Lagrange interpolation*

$$\frac{f(x) - P_n(x)}{f(x)},$$

for example, we must modify the algorithm slightly. We demand

$$\frac{|f(x) - P_n(x)|}{|f(x)|} \leqslant \varepsilon \tag{7}$$

everywhere on $[a,b]$, which gives

$$|f(x) - P_n(x)| \leqslant \varepsilon |f(x)|.$$

This is nothing more than absolute error control, with the desired tolerance now given by $\varepsilon |f(x)|$. Since $|f(x)|$ varies with x, we demand that

$$|f(x) - P_n(x)| \leqslant \varepsilon \min_{[a,b]} |f(x)| \equiv \varepsilon K$$

to ensure that (7) is satisfied everywhere on $[a, b]$. Hence, we find

$$N = \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_L S_L}{(n+1)! \varepsilon K} \right)^{\frac{1}{n+1}} \right\rceil .$$

A potential problem is immediately apparent: it is entirely possible that $K = 0$, in which case N is undetermined. If we define N_A to be the number of subintervals required for absolute error control, and N_R as the analogous quantity for relative error control, we have

$$N_A \equiv \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_L S_L}{(n+1)! \varepsilon} \right)^{\frac{1}{n+1}} \right\rceil$$
$$N_R \equiv \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_L S_L}{(n+1)! \varepsilon K} \right)^{\frac{1}{n+1}} \right\rceil .$$

Hence, when $K < 1, N_A < N_R$ and when $K > 1, N_R < N_A$. This provides a way of choosing between relative and absolute error control - we choose that which requires the fewest subintervals, since the efficiency of the interpolating algorithm is determined by the total number of nodes required in the construction of the piecewise polynomial. Note that when $K = 1, N_R = N_A$.

The same reasoning holds for Hermite interpolation, as in

$$N_A \equiv \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_H S_H}{(2n+2)! \varepsilon} \right)^{\frac{1}{2n+2}} \right\rceil$$
$$N_R \equiv \left\lceil \left(\frac{b-a}{n} \right) \left(\frac{M_H S_H}{(2n+2)! \varepsilon K} \right)^{\frac{1}{2n+2}} \right\rceil .$$

8 Exercises

1. Use the properties of the Vandermonde matrix to show that, if the interpolation nodes are distinct, then both the the Lagrange coefficient polynomials and the Lagrange interpolating polynomial exist and are unique.

2. Use the fact that the Lagrange coefficient polynomials exist, together with the structural definition of the Hermite interpolating polynomial, to deduce that the Hermite interpolating polynomial exists, provided that the interpolation nodes are distinct.

3. Show that the Hermite interpolating polynomial is unique.

4. Derive the error term for the Hermite interpolating polynomial.

5. Consider the task of approximating

$$f(x) = x^2 e^{2x}$$

on the interval $[0, 3]$, by means of a piecewise Hermite interpolating polynomial of degree three. Assuming the nodes are equispaced, what is the spacing h of the nodes required for an absolute tolerance of $\varepsilon = 10^{-8}$?

6. Consider the task of approximating

$$f(x) = x^3 e^{x}$$

on the interval $[-1, 4]$, by means of a piecewise Lagrange interpolating polynomial of degree four. Assuming the nodes are equispaced, what is the spacing h of the nodes required for an absolute tolerance of $\varepsilon = 10^{-7}$?

7. Consider the task of approximating Runge's function

$$R(x) = \frac{1}{1 + x^2}$$

on the interval $[-5, 5]$ by means of a piecewise cubic Lagrange interpolating polynomial. Show that the magnitude of the resulting absolute approximation error is bounded by

$$36h^4,$$

where h is the uniform node spacing on $[-5, 5]$.

9 Solutions

1. Assume that the nodes $\{x_0, x_1, \ldots, x_n\}$ are distinct. The Lagrange coefficient polynomial satisfies

$$L_{n,k}(x_j) = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}.$$

If we assume that $L_{n,k}(x)$ has the form

$$L_{n,k}(x) = \sum_{i=0}^{n} l_i x^i$$

then we have

$$L_{n,k}(x_j) = \sum_{i=0}^{n} l_i x_j^i = \delta_{jk},$$

which gives the linear system

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^n \\ \vdots & & & \ddots & \\ 1 & x_n & x_n^2 & & x_n^n \end{bmatrix} \begin{bmatrix} l_0 \\ l_1 \\ \vdots \\ l_n \end{bmatrix} = \mathbf{e}_{k+1}.$$

The coeffficient matrix on the LHS is known as a *Vandermonde* matrix and, if the nodes are distinct, the Vandermonde matrix is nonsingular. This means that this system is invertible, and so its solution exists and is unique. Hence, $L_{n,k}(x)$ exists and is unique. If we assume that the Lagrange interpolating polynomial has the form

$$P_n(x) = \sum_{i=0}^{n} a_i x^i$$

then similar reasoning gives

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^n \\ \vdots & & & \ddots & \\ 1 & x_n & x_n^2 & & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}.$$

Again, the invertibility of the Vandermonde matrix ensures that $P_n(x)$ exists and is unique.

2. We observe that the structural definiton of $H_{2n+1}(x)$ is given in terms of $L_{n,k}(x)$. Since $L_{n,k}(x)$ exists (for distinct nodes), so does $L_{n,k}^2(x)$ and $L'_{n,k}(x_k)$. Of course, $(x - x_k)$ exists, and so we conclude that $H_{2n+1}(x)$ exists (for distinct nodes).

3. Say $P(x)$ is a polynomial of degree at most $2n+1$ that satisfies the same properties as $H_{2n+1}(x)$ but such that $P(x) \neq H_{2n+1}(x)$. Define

$$D(x) \equiv H_{2n+1}(x) - P(x)$$

which means that $D(x)$ has degree at most $2n+1$. But $D(x_k) = 0$ and $D'(x_k) = 0$ for each $k = 0, ..., n$ which means that $D(x)$ has a zero of multiplicity two at each x_k. Hence,

$$D(x) = (x - x_0)^2 (x - x_1)^2 ... (x - x_n)^2 Q(x)$$

where $Q(x)$ is some polynomial. But this implies that $D(x)$ has degree at least $2n+2$ if $Q(x)$ is nonzero. This is untenable, and so $Q(x) = 0$, which means $D(x) = 0$, and so

$$P(x) = H_{2n+1}(x).$$

4. Assume the interval of approximation is $[a, b]$ and define

$$g(t) \equiv f(t) - H_{2n+1}(t) - \frac{(t - x_0)^2 ... (t - x_n)^2}{(x - x_0)^2 ... (x - x_n)^2} [f(x) - H_{2n+1}(x)].$$

Now

$$g(x_k) = 0 \qquad k = 0, 1, ..., n$$

and

$$g(x) = 0$$

for any $x \neq x_k, x \in [a, b]$. So $g(t)$ has $n+2$ distinct roots in $[a, b]$. By Rolle's theorem, then, $g'(t)$ has $n+1$ distinct roots $\{\xi_0, ..., \xi_n\}$ located between the nodes $\{x_0, ..., x_n, x\}$. Also, $g'(x_k) = 0$ for each $k = 0, ..., n$. Hence, $g'(t)$ has $n+1$ distinct roots at $\{x_0, ..., x_n\}$ and so has a total of $2n+2$ distinct roots at $\{x_0, ..., x_n, x, \xi_0, ..., \xi_n\}$. Now $g'(t)$ is $2n+1$ times differentiable (since $f \in C^{2n+2}[a, b]$), so the Generalized Rolle's Theorem ensures that there exists $\xi \in [a, b]$ such that $g^{(2n+2)}(\xi) = 0$. Hence, we have,

$$g^{(2n+2)}(t) = f^{(2n+2)}(t) - H_{2n+1}^{(2n+2)}(t) - \frac{(2n+2)!}{(x - x_0)^2 ... (x - x_n)^2} [f(x) - H_{2n+1}(x)]$$

$$= f^{(2n+2)}(t) - \frac{(2n+2)!}{(x - x_0)^2 ... (x - x_n)^2} [f(x) - H_{2n+1}(x)]$$

so that

$$g^{(2n+2)}(\xi) = 0$$
$$\Rightarrow [f(x) - H_{2n+1}(x)] = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x-x_0)^2 \ldots (x-x_n)^2.$$

Moreover, this error formula also holds for each $x = x_k$.

5. For cubic Hermite interpolation ($n = 1$), the pointwise error is bounded by

$$|\Delta(x)| \leqslant \frac{\max\left|f^{(4)}(x)\right|}{4!}\left|(x-x_0)^2(x-x_1)^2\right|.$$

For the given function $f(x) = x^2e^{2x}$, we have

$$f^{(4)}(x) = \left(48 + 64x + 16x^2\right)e^{2x}.$$

Clearly, this is an increasing function of x on $[0,3]$, and so achieves its maximum magnitude at $x = 3$. So

$$\max_{[0,3]}\left|f^{(4)}(x)\right| = 154916.66.$$

With the substitutions

$$x = x_0 + sh, x_i = x_0 + ih, s \in [0,1]$$

we have

$$(x-x_0)^2(x-x_1)^2 = h^4(s)^2(s-1)^2,$$

which gives

$$|\Delta(x)| \leqslant \frac{h^4 \max\limits_{[0,3]}\left|f^{(4)}(x)\right| \max\limits_{[0,1]}\left|(s)^2(s-1)^2\right|}{24}.$$

To find $\max\limits_{[0,1]}\left|(s)^2(s-1)^2\right|$ we consider

$$\frac{d(s)^2(s-1)^2}{ds} = 0 \Rightarrow 2s(s-1)(2s-1) = 0$$
$$\Rightarrow s = 0, s = 1, s = \frac{1}{2}.$$

These and the endpoints $s = \{0,1\}$ give

$$\max_{[0,1]}\left|(s)^2(s-1)^2\right| = \left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^2 = \frac{1}{16}.$$

Hence,

$$|\Delta(x)| \leqslant \frac{154916.66h^4}{384} \leqslant 10^{-8}$$
$$\Rightarrow h \leqslant 0.0022313.$$

If N is the number of subintervals on $[0,3]$, each of length h, we have

$$N \geqslant \frac{3}{h} = 1344.5.$$

We choose the smallest integer consistent with this inequality, i.e.

$$N = 1345,$$

which gives

$$h = \frac{3}{N} = 0.0022305.$$

6. For Lagrange interpolation of degree four ($n = 4$), the pointwise error is bounded by

$$|\Delta(x)| \leqslant \frac{\max\left|f^{(5)}(x)\right|}{5!}\left|\prod_{k=0}^{4}(x - x_k)\right|.$$

For the given function $f(x) = x^3e^x$, we have

$$f^{(5)}(x) = \left(60 + 60x + 15x^2 + x^3\right)e^x.$$

Computing $f^{(6)}(x)$ and setting $f^{(6)}(x) = 0$ yields three roots , none of which are on $[-1,4]$. Hence, $f^{(5)}(x)$ is an increasing function of x on $[-1,4]$, and so achieves its maximum magnitude at $x = 4$. This gives

$$\max_{[-1,4]}\left|f^{(5)}(x)\right| = 32977.28.$$

With the substitutions

$$x = x_0 + sh, x_i = x_0 + ih, s \in [0,4]$$

we have

$$\prod_{k=0}^{4}(x - x_k) = h^5\prod_{k=0}^{4}(s - k),$$

which gives

$$|\Delta(x)| \leqslant \frac{h^5 \max\limits_{[-1,4]} \left|f^{(5)}(x)\right| \max\limits_{[0,4]} \left|\prod\limits_{k=0}^{4}(s-k)\right|}{120}.$$

To find $\max\limits_{[0,4]} \left|\prod\limits_{k=0}^{4}(s-k)\right|$ we consider

$$\frac{d\left(\prod\limits_{k=0}^{4}(s-k)\right)}{ds} = 0 \;\Rightarrow\; s = \begin{cases} 3.644 \\ 2.544 \\ 1.456 \\ 0.356 \end{cases}.$$

These, together with the endpoints $s = \{0, 4\}$, give

$$\max_{[0,4]} \left|\prod_{k=0}^{4}(s-k)\right| = 3.6314.$$

Hence,

$$N \geqslant \left(\frac{5}{4}\right)\left(\frac{(32977.28)(3.6314)}{(120)(10^{-7})}\right)^{\frac{1}{5}} = 124.9.$$

We choose the smallest integer consistent with this inequality, i.e.

$$N = 125,$$

which gives

$$h = \frac{b-a}{nN} = \frac{5}{500} = 0.01.$$

7. Since the polynomial is cubic, we have $n = 3$ and so we consider

$$R^{(4)}(x) = \frac{24\left(5x^4 - 10x^2 + 1\right)}{\left(1+x^2\right)^5}.$$

Hence,

$$R^{(5)}(x) = 0 \;\Rightarrow\; x\left(3x^4 - 10x^2 + 3\right) = 0$$

which has roots

$$x = 0$$
$$x = \pm\frac{1351}{780}$$
$$x = \pm\frac{780}{1351}.$$

These give

$$\max \left| R^{(4)}(x) \right| = 24,$$

which occurs at $x = 0$. (At the endpoints of the interval ($x = \pm 5$) we have $R^{(4)}(x) = 97/16697$).For cubic Lagrange interpolation, the pointwise error is bounded by

$$\begin{aligned} |\Delta(x)| &\leqslant \frac{\max \left| R^{(4)}(x) \right|}{4!} |(x - x_0)(x - x_1)(x - x_2)(x - x_3)| \\ &= \frac{24}{24} |(x - x_0)(x - x_1)(x - x_2)(x - x_3)|. \end{aligned}$$

With the substitutions

$$x = x_0 + sh, \ x_i = x_0 + ih, \ s \in [0, 3]$$

we have

$$\begin{aligned} |(x - x_0)(x - x_1)(x - x_2)(x - x_3)| &= \left| h^4 (s)(s - 1)(s - 2)(s - 3) \right| \\ &\leqslant h^4 (3)(2)(2)(3) \\ &= 36h^4. \end{aligned}$$

Hence,

$$|\Delta(x)| \leqslant 36h^4.$$

10 Continuous Least-Squares Approximation

Here, we define a measure of quality of an approximation by means of a certain type of integral. We then impose the condition that the approximating polynomial is determined through the minimization of this integral. So, if we wish to approximate $f(x)$ on $[a,b]$ with a polynomial $Q_n(x)$ of degree n, we demand that $Q_n(x)$ be such that

$$E(f,w,n,a,b)) \equiv \int_a^b w(x)\left(f(x)-Q_n(x)\right)^2 dx$$

is a minimum. In this expression, $w(x)$ is a nonnegative function on $[a,b]$, known as a *weight function*. Hence, the integrand is always positive. $E(f,w,n,a,b))$ represents an approximation error. Minimization of $E(f,w,n,a,b))$ is known as the *continuous least-squares problem*.

10.1 Bases for $\prod_n$

Let $\prod_n$ denote the space of polynomials of degree n or less with real coefficients. There are two bases for $\prod_n$ which are of interest here: the set of *monomials*

$$\{1, x, \ldots, x^n\},$$

and the set of *w-orthogonal polynomials*

$$\{\phi_0, \phi_1, \ldots, \phi_n\},$$

defined by the property

$$\int_a^b w(x)\,\phi_i(x)\,\phi_j(x)\,dx = \begin{cases} 0 & i \neq j \\ \alpha_j > 0 & i = j \end{cases}.$$

10.2 Solution of the Continuous Least-Squares Problem

An obvious approach is to write $Q_n(x)$ in terms of the monomial basis, as in

$$Q_n(x) = \sum_{i=0}^{n} q_i x^i,$$

then differentiate $E(f,w,n,a,b))$ with respect to each coefficient q_i, and solve the resulting system of equations. The system to be solved is linear and is known as a Hilbert system. It is invertible, which means that $Q_n(x)$ exists and is unique. However, the Hilbert coefficient matrix is often badly conditioned and its numerical inverse can be unreliable.

There is a more elegant way to find $Q_n(x)$. If we write

$$Q_n(x) = \sum_{i=0}^{n} a_i \phi_i(x),$$

so that $Q_n(x)$ is now in terms of the w-orthogonal basis, we have

$$E(f,w,n,a,b)) \equiv \int_a^b w(x) \left(f(x) - \sum_{i=0}^{n} a_i \phi_i(x) \right)^2 dx.$$

Differentiating with respect to each a_j $(j = 0, 1, \ldots .n)$ and setting equal to zero gives

$$\begin{aligned}
&\frac{\partial E(f,w,n,a,b))}{\partial a_j} = 2 \int_a^b w(x) \left(f(x) - \sum_{i=0}^{n} a_i \phi_i(x) \right) \left(-\phi_j(x) \right) dx = 0 \\
\Rightarrow &\int_a^b w(x) f(x) \phi_j(x) dx = \int_a^b w(x) \sum_{i=0}^{n} a_i \phi_i(x) \phi_j(x) dx \\
\Rightarrow &\int_a^b w(x) f(x) \phi_j(x) dx = \sum_{i=0}^{n} a_i \left(\int_a^b w(x) \phi_i(x) \phi_j(x) dx \right).
\end{aligned} \tag{8}$$

The expression (8) indicates a system of n equations, known as the *normal* equations. Since the $\phi_i(x)$ are orthogonal, we have

$$\int_a^b w(x) f(x) \phi_j(x) dx = a_j \alpha_j$$

$$\Rightarrow a_j = \frac{1}{\alpha_j} \int_a^b w(x) f(x) \phi_j(x) dx.$$

This is a very practical way of determining $Q_n(x)$, and avoids the inversion of a Hilbert system. The integral, however, may be difficult to find analytically, in which case a numerical method (such as Gaussian quadrature) would have to be used.

10.3 The Gram-Schmidt Process

The w-orthogonal polynomials on a given interval $[a, b]$ can be found using the *Gram-Schmidt* process:

$$\phi_0(x) \equiv 1$$

$$\phi_1(x) = x - \frac{\int_a^b xw(x)\phi_0(x)\phi_0(x)\,dx}{\int_a^b w(x)\phi_0(x)\phi_0(x)\,dx}$$

and, for $k \geqslant 2$,

$$\phi_k(x) = (x - B_k)\phi_{k-1}(x) - C_k\phi_{k-2}(x)$$

where

$$B_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-1}(x)\,dx}{\int_a^b w(x)\phi_{k-1}(x)\phi_{k-1}(x)\,dx}$$

$$C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x)\,dx}{\int_a^b w(x)\phi_{k-2}(x)\phi_{k-2}(x)\,dx}.$$

These polynomials are orthogonal on $[a, b]$, with respect to the weight function $w(x)$.

10.4 The Legendre Case

A special case occurs when $w(x) = 1$. In this case the orthogonal polynomials are known as Legendre polynomials, denoted $P_i(x)$, and we have

$$\int_a^b w(x)\phi_i(x)\phi_j(x)\,dx = \int_a^b P_i(x)P_j(x)\,dx$$

$$Q_n(x) = \sum_{i=0}^{n} a_i P_i(x)$$

and

$$E(f, 1, n, a, b)) = \int_a^b (f(x) - Q_n(x))^2\,dx.$$

The term $f(x) - Q_n(x)$ is a pointwise error, so that this expression is manifestly a measure of the 'total' pointwise error in the approximation $Q_n(x)$. This case with $w(x) = 1$ is, practically speaking, the most usual case that we consider in this type of approximation.

11 Exercises

1. We have, on some interval $[a, b]$,

$$Q_n(x) = \sum_{i=0}^{n} q_i x^i = \sum_{i=0}^{n} a_i \phi_i(x),$$

where $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is a set of orthogonal polynomials on $[a, b]$. Show that there exists a well-defined and unique linear relationship between the coefficients q_i and a_i.

2. What are the roots of $Q_{n+1}(x) \neq 0$ on $[a, b]$ if the integral

$$\int_a^b [Q_{n+1}(x)]^2 \, dx$$

is minimized? Assume that $Q_{n+1}(x)$ is monic.

3. Show that

$$\int_a^b P_{n+1}(x) \, Q_n(x) \, dx = 0,$$

where $Q_n(x)$ is any polynomial of degree n.

4. Consider the integral

$$I \equiv \int_{-1}^{1} p^2(x) \, dx$$

$$p(x) \equiv \sum_{i=0}^{3} i c_i \phi_i(x),$$

where $\phi_i(x)$ is a polynomial of degree i, and c_i is real. By minimizing I in the least-squares sense, and imposing the condition that $p(x)$ should be a monic polynomial of degree three, show that

$$p(x) = x^3 - \frac{3}{5}x.$$

12 Solutions

1. Write

$$\begin{aligned}\phi_i(x) &= \sum_{j=0}^{n} b_j^i x^j \\ b_i^i &\neq 0 \\ b_j^i &= 0 \text{ for } j > i\end{aligned}$$

Hence,

$$\begin{aligned}\sum_{i=0}^{n} q_i x^i &= \sum_{i=0}^{n} a_i \sum_{j=0}^{n} b_j^i x^j = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i b_j^i x^j \\ &= \sum_{j=0}^{n}\sum_{i=0}^{n} a_j b_i^j x^i = \sum_{i=0}^{n}\sum_{j=0}^{n} a_j b_i^j x^i.\end{aligned}$$

In the last line, we have simply interchanged the indices i and j, since they are independent of each other, and we have interchanged the summation symbols. This gives

$$q_i = \sum_{j=0}^{n} a_j b_i^j.$$

The RHS of this expression is an element of the product of a matrix B with $B_{ij} = b_i^j$, and a column vector $\mathbf{a}$. In explicit form, we have

$$\begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} b_0^0 & b_0^1 & \cdots & b_0^n \\ 0 & b_1^1 & & b_1^n \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & b_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}.$$

The coefficient matrix B is upper triangular, with all diagonal entries nonzero. Hence, it is invertible. Thus, there exists a unique and well-defined relationship between the coefficients q_i and a_i.

2. Consider the task of minimizing

$$\int_a^b \left[Q_{n+1}(x)\right]^2 dx = \int_a^b \left(\prod_{i=0}^{n+1}(x - x_i)\right)^2 dx$$

where $Q_{n+1}(x) \neq 0$. The problem is of the type in which the integral

$$\int_a^b w(x) \left(f(x) - \sum_{i=0}^{n+1} a_i \phi_i(x) \right)^2 dx$$

must be minimized. For the given problem we identify

$$w(x) = 1 \qquad f(x) = 0 \qquad Q_{n+1}(x) = \sum_{i=0}^{n+1} a_i P_i(x)$$

where the P_i are w-orthogonal polynomials on $[a, b]$. So the P_i are Legendre polynomials, since $w(x) = 1$. Clearly, $Q_{n+1}(x)$ is a linear combination of polynomials of varying degree (up to $n+1$), and so we must certainly have $a_{n+1} \neq 0$. Now

$$\int_a^b \left(\sum_{i=0}^{n+1} a_i P_i(x) \right)^2 dx = \int_a^b \left(a_0^2 (P_0)^2 + ... + a_{n+1}^2 (P_{n+1})^2 + \text{ other cross terms} \right) dx$$

where the "other cross terms" have the form $P_i P_j$ with $i \neq j$. Say

$$\alpha_j \equiv \int_a^b (P_j)^2 dx$$

which means that $\alpha_j > 0$ since $(P_j)^2 > 0$. Then we have

$$\int_a^b \left(a_0^2 (P_0)^2 + ... + a_{n+1}^2 (P_{n+1})^2 + \text{ other cross terms} \right) dx = a_0^2 \alpha_0 + ... + a_{n+1}^2 \alpha_{n+1}$$

since $\int_a^b P_i P_j dx = 0$ when $i \neq j$. Each term in $a_0^2 \alpha_0 + ... + a_{n+1}^2 \alpha_{n+1}$ is positive, and we must have $a_{n+1} \neq 0$, so this sum is minimized only if $a_i = 0$ for $i = 0, 1, ..., n$. This gives

$$Q_{n+1}(x) = a_{n+1} P_{n+1}(x)$$

and a_{n+1} would be chosen so that $Q_{n+1}(x)$ is monic on $[a, b]$. So, in order for the integral to be minimized, subject to the condition $Q_{n+1}(x) \neq 0$, we find that $Q_{n+1}(x)$ must be the monic Legendre polynomial of degree $n+1$ on $[a, b]$. Hence, its roots x_i are the roots of $P_{n+1}(x)$ on $[a, b]$.

3. Let the set $\{P_0, P_1, \ldots, P_n\}$ denote the Legendre polynomials on $[a, b]$, as determined from the Gram-Schmidt process, for example. This set is a basis for $\prod_n$. Hence,

$$Q_n = \sum_{i=0}^{n} a_i P_i(x)$$

and so

$$\begin{aligned}\int_a^b P_{n+1}(x) Q_n(x)\, dx &= \int_a^b P_{n+1}(x) \sum_{i=0}^{n} a_i P_i(x)\, dx \\ &= \sum_{i=0}^{n} a_i \int_a^b P_{n+1}(x) P_i(x)\, dx \\ &= 0,\end{aligned}$$

since the Legendre polynomials are orthogonal.

4. The problem is of the type in which the integral

$$\int_a^b w(x) \left(g(x) - \sum_{i=0}^{n} i c_i \phi_i(x) \right)^2 dx$$

must be minimized (the continuous least-squares problem). For the given problem we identify

$$w(x) = 1 \qquad g(x) = 0 \qquad p(x) = \sum_{i=0}^{3} i c_i \phi_i(x)$$

where the $\phi_i(x)$ are polynomials of degree i, orthogonal with respect to $w(x) = 1$. In other words, the $\phi_i(x)$ are Legendre polynomials. We know that $p(x)$ must be a monic polynomial of degree three. Clearly, $p(x)$ is a linear combination of Legendre polynomials and so we must have $c_3 \neq 0$. Now

$$\int_{-1}^{1} \left(\sum_{i=0}^{3} i c_i \phi_i(x) \right)^2 dx = \int_{-1}^{1} \left(c_1^2 \phi_1^2 + 4 c_2^2 \phi_2^2 + 9 c_3^2 \phi_3^2 + \text{ other cross terms} \right) dx$$

where the "other cross terms" have the form $ijc_ic_j\phi_i\phi_j$ with $i \neq j$. Say

$$\alpha_j \equiv \int_{-1}^{1} \phi_j^2 dx$$

which means that $\alpha_j > 0$ since $\phi_j^2 > 0$. Then we have

$$\int_{-1}^{1} \left(c_1^2\phi_1^2 + 4c_2^2\phi_2^2 + 9c_3^2\phi_3^2 + \text{ other cross terms}\right) dx = c_1^2\alpha_1 + 4c_2^2\alpha_2 + 9c_3^2\alpha_3$$

since $\int_{-1}^{1} \phi_i\phi_j dx = 0$ when $i \neq j$. Each term in $c_1^2\alpha_1 + 4c_2^2\alpha_2 + 9c_3^2\alpha_3$ is positive and we must have $c_3 \neq 0$, so this sum is minimized only if $c_i = 0$ for $i = 1, 2$. This gives

$$p(x) = 3c_3\phi_3(x),$$

where $\phi_3(x) = x^3 - \frac{3}{5}x$ (from the Gram-Scmidt process) and $c_3 = \frac{1}{3}$, so that $p(x)$ is monic on $[-1, 1]$. In other words, $p(x)$ is the monic Legendre polynomial of degree three on $[-1, 1]$. Such polynomial is given by

$$p(x) = (3)\left(\frac{1}{3}\right)\left(x^3 - \frac{3}{5}x\right) = x^3 - \frac{3}{5}x.$$

13 Quadrature

13.1 Notation and Terminology

Quadrature refers to the numerical evaluation of definite integrals. A quadrature *rule* or *formula* is typically a linear combination of values of the integrand sampled at a finite set of nodes. So, we write

$$Q\left[f,a,b,n+1\right] \equiv \sum_{i=0}^{n} c_i f\left(x_i\right)$$

for an $(n+1)$-point quadrature rule, where the nodes $x_i \in [a,b]$. If a and b are nodes, we say the quadrature rule is *closed;* if the nodes are strictly within $[a,b]$, we say the quadrature rule is *open*. The coefficients c_i are known as *weights*.

We denote the integral by

$$I\left[f,a,b\right] \equiv \int_a^b f\left(x\right) dx$$

and, since $Q\left[f,a,b,n+1\right]$ is intended as an approximation to the integral, we have, in general,

$$Q\left[f,a,b,n+1\right] \approx I\left[f,a,b\right].$$

13.2 Interpolatory Quadrature

If the weights in $Q\left[f,a,b,n+1\right]$ are given by the expression

$$c_i = \int_a^b L_{n,i}\left(x\right) dx = \int_a^b \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{\left(x-x_k\right)}{\left(x_i-x_k\right)} dx$$

then the quadrature rule is termed *interpolatory* quadrature. This definition arises from the integration of the Lagrange interpolating polynomial

$$\begin{aligned}\int_a^b P_n(x)\,dx &= \int_a^b \sum_{i=0}^{n} L_{n,i}(x)\,f(x_i)\,dx \\ &= \sum_{i=0}^{n}\left(\int_a^b L_{n,i}(x)\right) f(x_i)\,dx \\ &= \sum_{i=0}^{n} c_i f(x_i)\end{aligned}$$

and, since $P_n(x)$ approximates $f(x)$, $\int_a^b P_n(x)\,dx$ is taken as an approximation to $\int_a^b f(x)\,dx$.

13.3 Newton-Cotes Quadrature

If the nodes are equispaced, with spacing h, then the interpolatory quadrature rules that emerge are known as *Newton-Cotes* quadrature rules. Two well-known examples of Newton-Cotes quadrature are the Trapezium Rule

$$Q[f,a,b,2] = \frac{h}{2}f(x_0) + \frac{h}{2}f(x_1),$$

where $h = x_1 - x_0$ and $c_0 = c_1 = \frac{h}{2}$, and Simpson's Rule

$$Q[f,a,b,3] = \frac{h}{3}f(x_0) + \frac{4h}{3}f(x_1) + \frac{h}{3}f(x_2),$$

where $h = \frac{(x_2 - x_0)}{2}$, $c_0 = c_2 = \frac{h}{3}$ and $c_1 = \frac{4h}{3}$.

The approximation error for general Newton-Cotes rules is given by

$$\frac{h^{n+3}f^{(n+2)}(\vartheta)}{(n+2)!}\int_0^n t^2(t-1)\ldots(t-n)\,dt \qquad \text{for } n \text{ even}$$

$$\frac{h^{n+2}f^{(n+1)}(\vartheta)}{(n+1)!}\int_0^n t(t-1)\ldots(t-n)\,dt \qquad \text{for } n \text{ odd}$$

where $\vartheta \in (a,b)$. The Trapezium Rule has $n = 1$, and Simpson's Rule has $n = 2$.

13.4 Degree of Precision of a Quadrature Rule

If a quadrature rule $Q[f, a, b, n+1]$ has *degree of precision* m then we have

$$Q\left[x^k, a, b, n+1\right] = I\left[x^k, a, b\right]$$

for $k = 0, 1, \ldots, m$. In other words, $Q[f, a, b, n+1]$ is exact for all monomials of degree up to and including m. A consequence of this is that, if $p_m(x)$ is an arbitrary polynomial of degree m, then

$$Q[p_m, a, b, n+1] = I[p_m, a, b].$$

Note that the Trapezium rule has degree of precision one, and Simpson's Rule has degree of precision three.

There is a correlation between the order of interpolatory quadrature and degree of precision. Consider the error formula for Newton-Cotes rules with even n. Since the derivative is of order $n+2$, the error vanishes for all monomials of degree $n+1$ or less. Hence, the rule has degree of precision $n+1$. But the order of the rule is $n+3$. So we have, in this case,

$$\text{order} \;=\; \text{degree of precision} + 2.$$

This is also true for the Newton-Cotes rules with odd n, although those rules have degree of precision n. Clearly, for these interpolatory rules, the order increases as the degree of precision increases. This important point informs the construction of Gaussian quadrature rules, where we attempt to maximize the degree of precision.

14 Exercises

1. Determine the Newton-Cotes quadrature rule with $n = 3$, including error term.

2. Determine the degree of precision of

$$Q[f, -1, 1, 2] \equiv f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right).$$

3. Determine the parameters c_0, c_1 and c_2 so that

$$Q[f, -1, 1, 3] \equiv c_0 f(-1) + c_1 f(0) + c_2 f(1)$$

has degree of precision two (at least).

4. Determine the parameters c_0, c_1 and x_1 so that

$$Q[f, 0, 1, 2] \equiv c_0 f(0) + c_1 f(x_1)$$

has degree of precision two. Verify that this rule does not have degree of precision three.

5. Determine the parameters c_1, x_0 and x_1 so that

$$Q[f, 0, 1, 2] \equiv \frac{1}{2} f(x_0) + c_1 f(x_1)$$

has degree of precision two. Does this rule have degree of precision three?

6. Determine the parameters a, b, c and d so that

$$Q[f, -1, 1, 2] \equiv af(-1) + bf(1) + cf'(-1) + df'(1)$$

has degree of precision three. Verify that this rule does not have degree of precision four.

15 Solutions

1. We have, with $n = 3$,

$$\int_{x_0}^{x_3} f(x)\,dx = \sum_{i=0}^{3} a_i f(x_i) + \frac{h^5 f^{(4)}(\vartheta)}{4!} \int_0^3 t(t-1)(t-2)(t-3)\,dt$$

where

$$a_i = \int_{x_0}^{x_3} \prod_{\substack{j=0 \\ j\neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right) dx.$$

With the substitutions

$$x_j = x_0 + jh, \quad x = x_0 + sh, \quad dx = hds$$

we have

$$a_0 = h\int_0^3 \left(\frac{s-1}{0-1}\right)\left(\frac{s-2}{0-2}\right)\left(\frac{s-3}{0-3}\right) ds = \frac{3h}{8}$$

$$a_1 = h\int_0^3 \left(\frac{s-0}{1-0}\right)\left(\frac{s-2}{1-2}\right)\left(\frac{s-3}{1-3}\right) ds = \frac{9h}{8}$$

$$a_2 = h\int_0^3 \left(\frac{s-0}{2-0}\right)\left(\frac{s-1}{2-1}\right)\left(\frac{s-3}{2-3}\right) ds = \frac{9h}{8}$$

$$a_3 = h\int_0^3 \left(\frac{s-0}{3-0}\right)\left(\frac{s-1}{3-1}\right)\left(\frac{s-2}{3-2}\right) ds = \frac{3h}{8}.$$

Also,

$$\int_0^3 t(t-1)(t-2)(t-3)\,dt = -\frac{9}{10}$$

and so

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right] - \frac{3h^5 f^{(4)}(\vartheta)}{80}.$$

2. We have

$$Q\left[f,-1,1,2\right]\equiv f\left(-\frac{\sqrt{3}}{3}\right)+f\left(\frac{\sqrt{3}}{3}\right).$$

Hence,

$f\left(x\right)$	$\int_{-1}^{1} f\left(x\right)dx$	$Q\left[f,-1,1,2\right]$
1	2	2
x	0	0
x^2	$\frac{2}{3}$	$\frac{2}{3}$
x^3	0	0
x^4	$\frac{2}{5}$	$\frac{2}{9}$

so that the degree of precision of $Q\left[f,-1,1,2\right]$ is 3.

3. We have

$$Q\left[f,-1,1,3\right]\equiv c_0f\left(-1\right)+c_1f\left(0\right)+c_2f\left(1\right).$$

Hence,

$f\left(x\right)$	$\int_{-1}^{1} f\left(x\right)dx$	$Q\left[f,-1,1,3\right]$
1	2	$c_0+c_1+c_2$
x	0	$-c_0+c_2$
x^2	$\frac{2}{3}$	c_0+c_2

which yields the linear system

$$\left[\begin{array}{ccc}1&1&1\\-1&0&1\\1&0&1\end{array}\right]\left[\begin{array}{c}c_0\\c_1\\c_2\end{array}\right]=\left[\begin{array}{c}2\\0\\\frac{2}{3}\end{array}\right]\Rightarrow\left[\begin{array}{c}c_0\\c_1\\c_2\end{array}\right]=\left[\begin{array}{c}\frac{1}{3}\\\frac{4}{3}\\\frac{1}{3}\end{array}\right]$$

which is, as expected, Simpson's rule.

4. We have

$$Q\left[f,0,1,2\right]\equiv c_0f\left(0\right)+c_1f\left(x_1\right).$$

Hence,

$f\left(x\right)$	$\int_{0}^{1} f\left(x\right)dx$	$Q\left[f,0,1,2\right]$
1	1	c_0+c_1
x	$\frac{1}{2}$	c_1x_1
x^2	$\frac{1}{3}$	$c_1x_1^2$

Hence,

$$x_1 = \frac{c_1 x_1^2}{c_1 x_1} = \frac{2}{3}$$
$$c_1 = \left(\frac{1}{x_1}\right)\frac{1}{2} = \frac{3}{4}$$
$$c_0 = 1 - c_1 = \frac{1}{4}$$

so that

$$Q\left[f, 0, 1, 2\right] = \frac{f\left(0\right)}{4} + \frac{3f\left(\frac{2}{3}\right)}{4}.$$

Furthermore,

$$\int_0^1 x^3 dx = \frac{1}{4}$$
$$Q\left[x^3, 0, 1, 2\right] = \frac{2}{9} \neq \frac{1}{4}$$

confirming that $Q\left[f, 0, 1, 2\right]$ does indeed have degree of precision two.

5. We have

$$Q\left[f, 0, 1, 2\right] \equiv \frac{1}{2} f\left(x_0\right) + c_1 f\left(x_1\right).$$

Hence,

$f\left(x\right)$	$\int_0^1 f\left(x\right) dx$	$Q\left[f, 0, 1, 2\right]$
1	1	$\frac{1}{2} + c_1$
x	$\frac{1}{2}$	$\frac{x_0}{2} + c_1 x_1$
x^2	$\frac{1}{3}$	$\frac{x_0^2}{2} + c_1 x_1^2$

Hence,

$$c_1 = 1 - \frac{1}{2} = \frac{1}{2}$$

so that

$$\frac{x_0}{2} + c_1 x_1 = \frac{x_0}{2} + \frac{x_1}{2} = \frac{1}{2}$$
$$\frac{x_0^2}{2} + c_1 x_1^2 = \frac{x_0^2}{2} + \frac{x_1^2}{2} = \frac{1}{3}$$

which give

$$x_0 + x_1 = 1 \;\Rightarrow\; x_0 = 1 - x_1$$
$$x_0^2 + x_1^2 = \frac{2}{3}.$$

Hence,

$$x_0^2 - x_0 + \frac{1}{6} = 0 \;\Rightarrow\; x_0 = \frac{1}{2} + \frac{1}{2\sqrt{3}} \;\text{ or }\; \frac{1}{2} - \frac{1}{2\sqrt{3}}$$

so that

$$x_1 = \frac{1}{2} - \frac{1}{2\sqrt{3}} \;\text{ or }\; \frac{1}{2} + \frac{1}{2\sqrt{3}}.$$

This all gives

$$Q\left[f, 0, 1, 2\right] = \frac{1}{2} f\left(\frac{1}{2} - \frac{1}{2\sqrt{3}}\right) + \frac{1}{2} f\left(\frac{1}{2} + \frac{1}{2\sqrt{3}}\right).$$

Furthermore,

$$\int_0^1 x^3 dx = \frac{1}{4} = Q\left[x^3, 0, 1, 2\right]$$
$$\int_0^1 x^4 dx = \frac{1}{5} \neq Q\left[x^4, 0, 1, 2\right] = 0.1944$$

confirming that $Q\left[f, 0, 1, 2\right]$ does indeed have degree of precision three.

6. We have

$$Q\left[f, -1, 1, 2\right] \equiv af\left(-1\right) + bf\left(1\right) + cf'\left(-1\right) + df'\left(1\right).$$

Hence,

$f\left(x\right)$	$\int_{-1}^{1} f\left(x\right) dx$	$Q\left[f, -1, 1, 2\right]$
1	2	$a + b$
x	0	$-a + b + c + d$
x^2	$\frac{2}{3}$	$a + b - 2c + 2d$
x^3	0	$-a + b + 3c + 3d$

which yields the linear system

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 2 \\ -1 & 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}.$$

Note that

$$\int_{-1}^{1} x^4 dx = \frac{2}{5} \neq Q\left[x^4, -1, 1, 2\right] = -\frac{2}{3}$$

so that the degree of precision of $Q\left[f, -1, 1, 2\right]$ is indeed three.

16 Gaussian Quadrature

16.1 Maximizing the Degree of Precision

A *Gaussian quadrature rule* has the form

$$G\left[f,w,a,b,n+1\right]\equiv\sum_{i=0}^{n}c_{i}^{w}f\left(x_{i}^{w}\right),$$

where the weights c_i^w and nodes x_i^w are free parameters to be determined. A Gaussian rule is intended to approximate the integral

$$I\left[f,w,a,b\right]\equiv\int_{a}^{b}w\left(x\right)f\left(x\right)dx,$$

where $w(x)$ is a weight function. Since there are $n+1$ weights and $n+1$ nodes in the Gaussian rule, we need $2n+2$ independent conditions to determine these parameters. These conditions are specified by demanding that the rule has a degree of precision of $2n+1$. In other words, we must have

$$G\left[x^{k},w,a,b,n+1\right]=I\left[x^{k},w,a,b\right]$$

for each $k=0,1,\ldots,2n+1$, which gives $2n+2$ equations to be solved simultaneously. These equations are linear in the weights c_i^w, but nonlinear in the nodes x_i^w, making their solution difficult, particularly for large n.

16.2 Theorems on Gaussian Quadrature

The following result gives the relationship between the weights, nodes and degree of precision of a Gaussian quadrature rule:

The $(n+1)$-point quadrature rule

$$G\left[f,w,a,b,n+1\right]\equiv\sum_{i=0}^{n}c_{i}^{w}f\left(x_{i}^{w}\right)\approx\int_{a}^{b}w\left(x\right)f\left(x\right)dx$$

where $w(x)$ is a weight function on $[a,b]$, has degree of precision $2n+1$ if and only if the nodes $\{x_0^w, x_1^w, \ldots, x_n^w\}$ are the roots of the w-orthogonal

polynomial $P_{n+1}^{w}(x)$ on $[a,b]$, and the weights c_i^w are given by

$$c_i^w = \int_a^b w(x) \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j^w}{x_i - x_j^w} \right) dx.$$

The approximation error in $G[f,w,a,b,n+1]$ is given by

$$\int_a^b w(x) f(x)\, dx - G[f,w,a,b,n+1] = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!} \int_a^b w(x) \left(\prod_{i=0}^{n} (x - x_i^w) \right)^2 dx.$$

where $\zeta \in (a,b)$.

The most usual case that arises in quadrature applications is the case where $w(x) = 1$. In this case, we write

$$G[f,1,a,b,n+1] \equiv \sum_{i=0}^{n} c_i f(x_i) \approx \int_a^b f(x)\, dx$$

$$c_i = \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) dx$$

$$\int_a^b f(x)\, dx - G[f,1,a,b,n+1] = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!} \int_a^b \left(\prod_{i=0}^{n} (x - x_i) \right)^2 dx$$

and the nodes x_i are the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a,b]$. Gaussian quadrature with the unit weight function is known as *Gauss-Legendre* quadrature. Note that the expression for the weights c_i necessarily defines Gauss-Legendre quadrature as interpolatory.

16.3 Gaussian Quadrature in terms of a Stepsize

We consider here the specific case of Gauss-Legendre quadrature. The roots of $P_{n+1}(x)$ on $[a,b]$ are symmetrically distributed within $[a,b]$, but they are not equispaced and they are not located at a or b (so Gauss-Legendre quadrature is open). Nevertheless, it is possible to write Gauss-Legendre quadrature in terms of a stepsize, as was done for Newton-Cotes quadrature.

We define

$$h \equiv \frac{b-a}{n+2},$$

so that h is the average separation of the nodes on $[a,b]$. We may now write

$$G\left[f,1,a,b,n+1\right] = h\sum_{i=0}^{n} d_i f\left(x_i\right),$$

where

$$d_i \equiv \frac{\left(n+2\right)c_i}{b-a}.$$

Furthermore, with the substitution $x = a + sh$, where $s \in [0, n+2]$ is a continuous variable, and $x_i = a + \sigma_i h$, where σ_i is an appropriate constant, we find

$$\int_a^b \left(\prod_{i=0}^{n}\left(x-x_i\right)\right)^2 dx = h^{2n+3}\int_0^{n+2}\left(\prod_{i=0}^{n}\left(s-\sigma_i\right)\right)^2 ds$$

so that

$$\int_a^b f\left(x\right)dx - G\left[f,1,a,b,n+1\right] = \frac{h^{2n+3}f^{(2n+2)}\left(\zeta\right)}{(2n+2)!}\int_0^{n+2}\left(\prod_{i=0}^{n}\left(s-\sigma_i\right)\right)^2 ds,$$

from which we conclude that Gauss-Legendre quadrature is of order $2n+3$.

16.4 Error control using Composite Gauss-Legendre Quadrature

Now consider the task of approximating $I\left[f,1,a,b\right]$, subject to a tolerance of ε, using *composite* Gauss-Legendre quadrature. The idea is to subdivide $[a,b]$ into N subintervals of equal length, and to perform $(n+1)$-point Gauss-Legendre quadrature on each subinterval, with predetermined n. The sum of these N quadratures then approximates $I\left[f,1,a,b\right]$. Hence, it is necessary to find an appropriate length for these subintervals.

From the above, we have for the error on an arbitrary subinterval $[\alpha,\beta]$ of length $\left(n+2\right)h$

$$\left|\int_\alpha^\beta f\left(x\right)dx - G\left[f,1,\alpha,\beta,n+1\right]\right| \leqslant \frac{h^{2n+3}\max\limits_{[\alpha,\beta]}\left|f^{(2n+2)}\left(x\right)\right|}{(2n+2)!}\left|\Upsilon\left(n\right)\right|,$$

where

$$\Upsilon(n) \equiv \int_0^{n+2} \left(\prod_{i=0}^{n} (s - \sigma_i) \right)^2 ds.$$

For N such subintervals on $[a, b]$ the magnitude of the total error $E(n, N, a, b)$ on $[a, b]$ has the bound

$$|E(n, N, a, b)| \leqslant N \left(\frac{h^{2n+3} \max\limits_{[a,b]} \left| f^{(2n+2)}(x) \right|}{(2n+2)!} \right) |\Upsilon(n)|$$

$$= \left(\frac{(b-a) |\Upsilon(n)| \max\limits_{[a,b]} \left| f^{(2n+2)}(x) \right|}{(n+2)(2n+2)!} \right) h^{2n+2}$$

since $N(n+2)h = b - a$.

We determine h from

$$\left(\frac{(b-a) |\Upsilon(n)| \max\limits_{[a,b]} \left| f^{(2n+2)}(x) \right|}{(n+2)(2n+2)!} \right) h^{2n+2} \leqslant \varepsilon.$$

This gives the largest allowable value of h as

$$h = |Q(n)| \left(\frac{\varepsilon}{(b-a) \max\limits_{[a,b]} |f^{(2n+2)}(x)|} \right)^{\frac{1}{2n+2}},$$

where

$$Q(n) \equiv \left(\frac{(n+2)(2n+2)!}{|\Upsilon(n)|} \right)^{\frac{1}{2n+2}}.$$

We determine the number of subintervals

$$N = \left\lceil \frac{b-a}{h(n+2)} \right\rceil,$$

and then a new stepsize

$$h^* = \frac{b-a}{N(n+2)}.$$

The length of each subinterval on $[a, b]$ is now given by $(n+2)h^*$, and $(n+1)$-point Gauss-Legendre quadrature is performed on each subinterval. It can be shown that the coefficient $Q(n)$ is greater than 2.44 for $n = 0, 1, \ldots, 19$.

Although we have only considered Gauss-Legendre quadrature here, the same principles hold for Gaussian quadrature with any weight function.

16.5 Other Properties

Some other properties of Gaussian quadrature are worth mentioning. Firstly, the weights in Gaussian quadrature rules are always positive. Secondly, Gaussian quadrature is uniformly convergent, meaning that the error in Gaussian quadrature tends to zero as n tends to infinity. Finally, tables of weights and nodes for Gaussian quadrature are often given in the literature with reference to the interval $[-1,1]$. The nodes x_i on $[\alpha,\beta]$ are related to the nodes $\widetilde{x}_i$ on $[-1,1]$ by

$$x_i = \frac{1}{2}\left[(\beta-\alpha)\,\widetilde{x}_i + \beta + \alpha\right],$$

and the weights c_i on $[\alpha,\beta]$ are related to the weights $\widetilde{c}_i$ on $[-1,1]$ by

$$c_i = \left(\frac{\beta-\alpha}{2}\right)\widetilde{c}_i.$$

16.6 Hermite Quadrature

Consider the integral of the Hermite interpolating polynomial

$$\begin{aligned}\int_a^b H_{2n+1}(x)\,dx &= \sum_{k=0}^{n}\left(\int_a^b H_{n,k}(x)\,dx\right) f(x_k) + \sum_{k=0}^{n}\left(\int_a^b \widehat{H}_{n,k}(x)\,dx\right) f'(x_k)\\ &\equiv \sum_{k=0}^{n} C_k f(x_k) + \sum_{k=0}^{n} D_k f'(x_k) \qquad (9)\end{aligned}$$

and

$$\begin{aligned}\int_a^b f(x)\,dx - \int_a^b H_{2n+1}(x)\,dx &= \int_a^b \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\prod_{k=0}^{n}(x-x_k)^2\,dx\\ &= \frac{f^{(2n+2)}(\zeta)}{(2n+2)!}\int_a^b \prod_{k=0}^{n}(x-x_k)^2\,dx, \qquad (10)\end{aligned}$$

where $\zeta \in (a,b)$. The expression in (9) constitutes a *Hermite quadrature rule*, with associated error given in (10). The weights in Hermite quadrature are

the integrals of the Hermite coefficient polynomials

$$C_k = \int_a^b H_{n,k}(x)\,dx$$

$$D_k = \int_a^b \widehat{H}_{n,k}(x)\,dx.$$

Note that, if the nodes x_k are chosen as the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a,b]$, then the resulting Hermite quadrature is equivalent to Gauss-Legendre quadrature (in such a case, it can be shown that $C_k = c_k$ and $D_k = 0$).

17 Exercises

1. Use the Hermite interpolating polynomial $H_{2n+1}(x)$ to derive the error term for Gauss-Legendre quadrature on the interval $[-1, 1]$.

2. The degree of precision of an interpolatory quadrature rule is correlated with its accuracy - the higher the degree of precision, the more accurate the rule. Given that Gauss-Legendre quadrature is designed to have maximal degree of precision, is there a sense in which its error is minimal?

3. What is the least number of subintervals required when composite 2-point Gauss-Legendre quadrature is used to determine

$$\int_{-5}^{5} \frac{dx}{1+x^2}$$

subject to a tolerance of $\varepsilon = 10^{-10}$?

4. What is the least number of subintervals required when composite 3-point Gauss-Legendre quadrature is used to determine

$$\int_{0}^{\pi} \sin x dx$$

subject to a tolerance of $\varepsilon = 10^{-8}$?

5. Show that, in Hermite quadrature,

$$D_k = \int_a^b \widehat{H}_{n,k}(x)\, dx = \int_a^b (x - x_k)\, L_{n,k}^2(x)\, dx = 0$$

when the nodes x_k are the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a, b]$.

6. Show that, in Hermite quadrature,

$$C_k = \int_a^b H_{n,k}(x)\, dx = \int_a^b \left[1 - 2(x - x_k) L'_{n,k}(x_k)\right] L_{n,k}^2(x)\, dx = c_k$$

when the nodes x_k are the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a, b]$.

18 Solutions

1. By Hermite interpolation, a polynomial $H_{2n+1}(x)$ of degree at most $2n+1$ exists such that, for $i = 0, 1, ..., n$,

$$H_{2n+1}(\widetilde{x}_i) = f(\widetilde{x}_i) \qquad H'_{2n+1}(\widetilde{x}_i) = f'(\widetilde{x}_i) \qquad \widetilde{x}_i \in [-1, 1]$$

The error formula for Hermite interpolation is

$$f(x) - H_{2n+1}(x) = \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\left[\pi(x)\right]^2,$$

where

$$\pi(x) \equiv \prod_{i=0}^{n}(x - \widetilde{x}_i).$$

It follows that

$$\int_{-1}^{1} f(x)dx - \int_{-1}^{1} H_{2n+1}(x)dx = \int_{-1}^{1} \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\left[\pi(x)\right]^2 dx.$$

Now, $H_{2n+1}(x)$ is of degree at most $2n+1$, and Gauss-Legendre quadrature has degree of precision $2n+1$, so that

$$\int_{-1}^{1} H_{2n+1}(x)\,dx = \sum_{i=0}^{n} \widetilde{c}_i H_{2n+1}(\widetilde{x}_i) = \sum_{i=0}^{n} \widetilde{c}_i f(\widetilde{x}_i).$$

Furthermore, since $\left[\pi(x)\right]^2 \geqslant 0$ the Mean-Value Theorem for Integrals may be used to write

$$\int_{-1}^{1} \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\left[\pi(x)\right]^2 dx = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!}\int_{-1}^{1}\left[\pi(x)\right]^2 dx$$

for some $\zeta \in (-1, 1)$. The continuity of $f^{(2n+2)}(\xi(x))$ is inferred from the second equation above. Thus, we have

$$\int_{-1}^{1} f(x)dx - \sum_{i=0}^{n} \widetilde{c}_i f(\widetilde{x}_i) = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!}\int_{-1}^{1}\left(\prod_{i=0}^{n}(x - \widetilde{x}_i)\right)^2 dx.$$

2. Since the nodes x_i must be the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a,b]$, we have that

$$\prod_{i=0}^{n}(x-x_i)=P_{n+1}(x),$$

so that

$$\frac{f^{(2n+2)}(\zeta)}{(2n+2)!}\int_a^b\left(\prod_{i=0}^{n}(x-x_i)\right)^2dx=\frac{f^{(2n+2)}(\zeta)}{(2n+2)!}\int_a^b\left(P_{n+1}(x)\right)^2dx.$$

But $\int_a^b\left(P_{n+1}(x)\right)^2dx$ is a minimum (in the sense of the continuous least-squares problem), and so we conclude that the maximality of the degree of precision of Gauss-Legendre quadrature results in the minimization of the integral in its error expression.

3. We have $n=1$. Repeated differentiation of $f(x)=\frac{1}{1+x^2}$ gives

$$f^{(4)}(x)=\frac{24}{\left(1+x^2\right)^3}-\frac{288x^2}{\left(1+x^2\right)^4}+\frac{384x^4}{\left(1+x^2\right)^5}.$$

This function has its maximum magnitude on $[-5,5]$ at $x=0$, giving

$$\max_{[-5,5]}\left|f^{(4)}(x)\right|=24.$$

Hence,

$$h=2.44\left(\frac{10^{-10}}{(10)(24)}\right)^{\frac{1}{4}}=0.00196$$

and

$$N=\left\lceil\frac{10}{3h}\right\rceil=1701$$
$$h^*=\frac{10}{5103}=0.00196.$$

4. We have $n=2$. Repeated differentiation of $f(x)=\sin x$ gives

$$f^{(6)}(x)=-\sin x.$$

This function has its maximum magnitude on $[0,\pi]$ at $x=\frac{\pi}{2}$, giving

$$\max_{[0,\pi]}\left|f^{(6)}(x)\right|=1.$$

Hence,

$$h = 2.44 \left(\frac{10^{-8}}{(\pi)(1)} \right)^{\frac{1}{6}} = 0.09358$$

and

$$N = \left\lceil \frac{\pi}{4h} \right\rceil = 9$$
$$h^* = \frac{\pi}{36} = 0.08727.$$

5. We have

$$L_{n,k}(x) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)}$$

so that

$$\begin{aligned}(x - x_k) L_{n,k}(x) &= (x - x_k) \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} \\ &= \frac{\prod_{i=0}^{n} (x - x_i)}{\prod_{i=0, i \neq k}^{n} (x_k - x_i)} \\ &= K P_{n+1}(x),\end{aligned}$$

where

$$K \equiv \frac{1}{\prod_{i=0, i \neq k}^{n} (x_k - x_i)}.$$

Hence,

$$\begin{aligned}D_k &= \int_a^b (x - x_k) L_{n,k}^2(x)\, dx \\ &= K \int_a^b P_{n+1}(x) L_{n,k}(x)\, dx \\ &= 0,\end{aligned}$$

since $L_{n,k}(x)$ is of degree n.

6.

$$\begin{aligned} C_k &= \int_a^b H_{n,k}(x)\,dx \\ &= \int_a^b \left[1 - 2(x - x_k) L'_{n,k}(x_k)\right] L^2_{n,k}(x)\,dx \\ &= \int_a^b L^2_{n,k}(x)\,dx - 2L'_{n,k}(x_k) \int_a^b (x - x_k) L^2_{n,k}(x)\,dx \\ &= \int_a^b L^2_{n,k}(x)\,dx. \end{aligned}$$

Now, $L^2_{n,k}(x)$ is of degree $2n$ so that

$$\int_a^b L^2_{n,k}(x)\,dx = \sum_{i=0}^{n} c_i L^2_{n,k}(x_i)$$

since Gauss-Legendre quadrature has degree of precision $2n+1$. The nodes x_i are the roots of the Legendre polynomial $P_{n+1}(x)$ on $[a,b]$. So,

$$\begin{aligned} \int_a^b L^2_{n,k}(x)\,dx &= \sum_{i=0}^{n} c_i L^2_{n,k}(x_i) \\ &= c_0 L^2_{n,k}(x_0) + \ldots + c_k L^2_{n,k}(x_k) + \ldots + c_n L^2_{n,k}(x_n) \\ &= c_k \end{aligned}$$

since $L_{n,k}(x_i) = 0$ when $x_i \neq x_k$, and $L_{n,k}(x_k) = 1$. Hence,

$$C_k = \int_a^b H_{n,k}(x)\,dx = c_k.$$

19 Numerical Differentiation

19.1 Linear Taylor System

Say we have the discrete data $\{f(x_0), f(x_1), ..., f(x_n)\}$ and assume that $x_0 < x_1 < \cdots < x_n$. Let $x \in [x_0, x_n]$ and define $h_i \equiv x_i - x$. Then we have

$$\begin{aligned} f_i \equiv f(x_i) &= f(x + h_i) \\ &= f(x) + h_i f'(x) + \frac{h_i^2}{2!} f''(x) + \ldots + \frac{h_i^n}{n!} f^{(n)}(x) + R_i \end{aligned}$$

for each $i = 0, 1, \ldots, n$. Here, $R_i = O\left(h_i^{n+1}\right)$ denotes the Taylor residual term. This yields the linear system

$$\begin{aligned} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} &= \begin{bmatrix} 1 & h_0 & \frac{h_0^2}{2} & \cdots & \frac{h_0^n}{n!} \\ 1 & h_1 & \frac{h_1^2}{2} & & \frac{h_1^n}{n!} \\ \vdots & & & \ddots & \vdots \\ 1 & h_n & \frac{h_n^2}{2} & \cdots & \frac{h_n^n}{n!} \end{bmatrix} \begin{bmatrix} f(x) \\ f'(x) \\ \vdots \\ f^{(n)}(x) \end{bmatrix} + \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_n \end{bmatrix} \\ &\equiv \mathbf{A}_n \begin{bmatrix} f(x) \\ f'(x) \\ \vdots \\ f^{(n)}(x) \end{bmatrix} + \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_n \end{bmatrix} \end{aligned}$$

Thus

$$\begin{bmatrix} f(x) \\ f'(x) \\ \vdots \\ f^{(n)}(x) \end{bmatrix} = \mathbf{A}_n^{-1} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} - \mathbf{A}_n^{-1} \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_n \end{bmatrix}.$$

This gives

$$f^{(j)}(x) \approx \ (j+1)\,\text{th row of } \mathbf{A}_n^{-1} \ \times \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}$$

for $j = 0, 1, \ldots, n$. The truncation error in this estimate of $f^{(j)}(x)$ is given by

$$(j+1)\,\text{th row of } \mathbf{A}_n^{-1} \ \times \left(- \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_n \end{bmatrix} \right).$$

If a roundoff error μ_i exists in f_i, we write $f_i + \mu_i$ in place of f_i in the above, giving

$$\begin{bmatrix} f(x) \\ f'(x) \\ \vdots \\ f^{(n)}(x) \end{bmatrix} = \mathbf{A}_n^{-1} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix} - \mathbf{A}_n^{-1} \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_n \end{bmatrix} + \mathbf{A}_n^{-1} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}.$$

19.2 Invertibility of $\mathbf{A}_n$

We have

$$\mathbf{A}_n = \begin{bmatrix} 1 & h_0 & \frac{h_0^2}{2} & \cdots & \frac{h_0^n}{n!} \\ 1 & h_1 & \frac{h_1^2}{2} & & \frac{h_1^n}{n!} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 1 & h_n & \frac{h_n^2}{2} & \cdots & \frac{h_n^n}{n!} \end{bmatrix} = \begin{bmatrix} 1 & h_0 & h_0^2 & \cdots & h_0^n \\ 1 & h_1 & h_1^2 & & h_1^n \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 1 & h_n & h_n^2 & \cdots & h_n^n \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ \vdots & & \frac{1}{2} & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{n!} \end{bmatrix}.$$

The first matrix on the RHS is a Vandermonde matrix, and is invertible if the h_i are distinct. The other matrix is a diagonal matrix with all diagonal entries nonzero, and so is invertible. Hence, their product is invertible, provided that the h_i are distinct. However, we have assumed that the nodes are distinct, and so the h_i are indeed distinct. Hence, we conclude that $\mathbf{A}_n$ is invertible.

19.3 A Note Regarding Roundoff Error

A finite precision computing device typically implements the roundoff process in a *relative* sense, so that we should write

$$\widetilde{f}_i = f_i(1 + \epsilon) = f_i + f_i\epsilon,$$

where $\widetilde{f}_i$ is the device value of f_i, and where ϵ is a small number representing the precision of the device. Furthermore, the roundoff error usually exists within a range, rather than being a precise value for all cases, and may have either sign (or even be zero). Hence, we should actually write

$$\widetilde{f}_i \in f_i + [-|f_i\epsilon|, |f_i\epsilon|],$$

which gives

$$\mu_i \in [-|f_i\epsilon|, |f_i\epsilon|] \qquad |\mu_i| \leqslant |f_i\epsilon| \qquad \max_i |\mu_i| = |\epsilon| \max_i |f_i|.$$

A typical value for $|\epsilon|$ is $2^{-53} \approx 10^{-16}$. Note that, when $|f_i| \approx 1, |\mu_i| \approx |\epsilon|$.

20 Exercises

1. Consider three equispaced nodes $\{x_0, x_1, x_2\}$ and choose $x = x_1$. Obtain approximations for $f'(x)$ and $f''(x)$ and the truncation errors in each of these approximations. Assume that roundoff errors are present in $\{f(x_0), f(x_1), f(x_2)\}$ and that these roundoff errors are bounded by μ. Hence, obtain expressions for the roundoff error in the approximations for $f'(x)$ and $f''(x)$.

2. Consider two equispaced nodes $\{x_0, x_1\}$ and choose $x = x_0 + \frac{h}{3}$. Obtain an approximation for $f'(x)$ and the truncation error in this approximation. Assume that roundoff errors are present in $\{f(x_0), f(x_1)\}$ and that these roundoff errors are bounded by μ. Hence, obtain an expression for the roundoff error in the approximation for $f'(x)$.

3. Use Taylor expansions to derive an $O(h)$ five-point formula which approximates $f'''(x_0)$ using $f(x_0 - h)$, $f(x_0), f(x_0 + h)$, $f(x_0 + 2h)$, and $f(x_0 + 3h)$, such that the leading error term is $-5hf^{(4)}(x_0)$. State the coefficient of $f(x_0)$ in this formula. Determine an upper bound for the roundoff error, and hence determine an optimal value for h. HINT: Construct
$$Af(x_0 - h) + Bf(x_0 + h) + Cf(x_0 + 2h) + Df(x_0 + 3h)$$
and obtain a linear system for A, B, C, D. You may use the fact that
$$\begin{bmatrix} -1 & 1 & 2 & 3 \\ 1 & 1 & 4 & 9 \\ -1 & 1 & 8 & 27 \\ 1 & 1 & 16 & 81 \end{bmatrix}^{-1} = \frac{1}{24} \begin{bmatrix} -6 & 11 & -6 & 1 \\ 36 & 6 & -24 & 6 \\ -12 & 4 & 12 & -4 \\ 2 & -1 & -2 & 1 \end{bmatrix}.$$

4. Use Taylor expansions to derive an $O(h^2)$ five-point formula which approximates $f''(x_0)$ using $f(x_0 - 2h), f(x_0 - h)$, $f(x_0)$, $f(x_0 + h)$, and $f(x_0 + 2h)$, such that the leading error term is $-3h^2 f^{(4)}(x_0)$. State the coefficient of $f(x_0)$ in this formula. HINT: Construct
$$Af(x_0 - 2h) + Bf(x_0 - h) + Cf(x_0 + h) + Df(x_0 + 2h)$$
and obtain a linear system for A, B, C, D. You may use the fact that
$$\begin{bmatrix} -2 & -1 & 1 & 2 \\ 4 & 1 & 1 & 4 \\ -8 & -1 & 1 & 8 \\ 16 & 1 & 1 & 16 \end{bmatrix}^{-1} = \frac{1}{24} \begin{bmatrix} 2 & -1 & -2 & 1 \\ -16 & 16 & 4 & -4 \\ 16 & 16 & -4 & -4 \\ -2 & -1 & 2 & 1 \end{bmatrix}.$$

21 Solutions

1. Assume we have the data $\{f_0, f_1, f_2\}$ where the nodes $\{x_0, x_1, x_2\}$ are equispaced (spacing h). Choose $x = x_1$. Hence, we have

$$\begin{aligned} h_0 &= x_0 - x_1 = -h \\ h_1 &= x_1 - x_1 = 0 \\ h_2 &= x_2 - x_1 = h \end{aligned}$$

and so

$$\begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 1 & -h & \frac{h^2}{2} \\ 1 & 0 & 0 \\ 1 & h & \frac{h^2}{2} \end{bmatrix} \begin{bmatrix} f(x_1) \\ f'(x_1) \\ f''(x_1) \end{bmatrix} + \begin{bmatrix} \frac{-h^3}{6} f'''(x_1) + R_0 \\ 0 \\ \frac{h^3}{6} f'''(x_1) + R_2 \end{bmatrix}$$

where R_0 and R_2 are the fourth-order residual terms in the Taylor expansions of f_0 and f_2 about x_1. We find, using the above expression for $\mathbf{A}_2$,

$$\mathbf{A}_2^{-1} = \begin{bmatrix} 1 & -h & \frac{h^2}{2} \\ 1 & 0 & 0 \\ 1 & h & \frac{h^2}{2} \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} f(x_1) \\ f'(x_1) \\ f''(x_1) \end{bmatrix} \approx \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ \frac{f_2 - f_0}{2h} \\ \frac{f_2 + f_0 - 2f_1}{h^2} \end{bmatrix}$$

as expected, with error term

$$-\begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} \frac{-h^3 f'''(x_1)}{6} + R_0 \\ 0 \\ \frac{h^3 f'''(x_1)}{6} + R_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{-h^3 f'''(x_1)}{6h} - \frac{(R_2 - R_0)}{2h} = O(h^2 + h^3) \\ \frac{-(R_0 + R_2)}{h^2} = O(h^2) \end{bmatrix}.$$

If roundoff errors $\{\mu_0, \mu_1, \mu_2\}$ exist in $\{f_0, f_1, f_2\}$ then an estimate of resultant roundoff error is given by

$$\begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \frac{\mu_2 - \mu_0}{2h} \\ \frac{\mu_2 + \mu_0 - 2\mu_1}{h^2} \end{bmatrix} \leqslant \begin{bmatrix} \mu \\ \frac{\mu}{h} \\ \frac{4\mu}{h^2} \end{bmatrix}$$

where $\mu \equiv \max\{|\mu_0|, |\mu_1|, |\mu_2|\}$.

2. Assume we have the data $\{f_0, f_1\}$ where the nodes $\{x_0, x_1\}$ are equi-spaced (spacing h). Choose $x = x_0 + \frac{h}{3}$. Hence, we have

$$h_0 = x_0 - x = -\frac{h}{3}$$
$$h_1 = x_1 - x = \frac{2h}{3}$$

and so

$$\begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{h}{3} \\ 1 & \frac{2h}{3} \end{bmatrix} \begin{bmatrix} f(x) \\ f'(x) \end{bmatrix} + \begin{bmatrix} R_0 \\ R_1 \end{bmatrix}$$

where R_0 and R_1 are the second-order residual terms in the Taylor expansions of f_0 and f_1 about x. We find, using the above expression for $\mathbf{A}_1$,

$$\mathbf{A}_1^{-1} = \begin{bmatrix} 1 & -\frac{h}{3} \\ 1 & \frac{2h}{3} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{h} & \frac{1}{h} \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} f(x) \\ f'(x) \end{bmatrix} \approx \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{h} & \frac{1}{h} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} \frac{2f_0+f_1}{3} \\ \frac{f_1-f_0}{h} \end{bmatrix}$$

as expected, with error term

$$-\begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{h} & \frac{1}{h} \end{bmatrix} \begin{bmatrix} R_0 \\ R_1 \end{bmatrix} = \begin{bmatrix} \frac{-(2R_0+R_1)}{3} \\ \frac{R_0-R_1}{h} \end{bmatrix} = \begin{bmatrix} O(h^2) \\ O(h) \end{bmatrix}.$$

If roundoff errors $\{\mu_0, \mu_1\}$ exist in $\{f_0, f_1\}$ then an estimate of resultant roundoff error is given by

$$\begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{h} & \frac{1}{h} \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix} = \begin{bmatrix} \frac{2\mu_0+\mu_1}{3} \\ \frac{\mu_1-\mu_0}{h} \end{bmatrix} \leqslant \begin{bmatrix} \mu \\ \frac{2\mu}{h} \end{bmatrix}$$

where $\mu \equiv \max\{|\mu_0|, |\mu_1|\}$.

3. We have

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)$$
$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)$$
$$f(x_0 + 2h) = f(x_0) + 2hf'(x_0) + \frac{4h^2}{2}f''(x_0) + \frac{8h^3}{6}f'''(x_0) + \frac{16h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)$$
$$f(x_0 + 3h) = f(x_0) + 3hf'(x_0) + \frac{9h^2}{2}f''(x_0) + \frac{27h^3}{6}f'''(x_0) + \frac{81h^4}{24}f^{(4)}(x_0) + O\left(h^5\right).$$

Now

$$\begin{aligned}
&Af(x_0-h)+Bf(x_0+h)+Cf(x_0+2h)+Df(x_0+3h)\\
&=(A+B+C+D)\,f(x_0)+(-A+B+2C+3D)\,hf'(x_0)\\
&\quad+(A+B+4C+9D)\,\frac{h^2}{2}f''(x_0)+(-A+B+8C+27D)\,\frac{h^3}{6}f'''(x_0)\\
&\quad+(A+B+16C+81D)\,\frac{h^4}{24}f^{(4)}(x_0)+O(h^5).
\end{aligned}$$

We require

$$\begin{aligned}
-A+B+2C+3D&=0\\
A+B+4C+9D&=0\\
-A+B+8C+27D&=6\\
A+B+16C+81D&=120,
\end{aligned}$$

which gives

$$\begin{bmatrix} -1 & 1 & 2 & 3\\ 1 & 1 & 4 & 9\\ -1 & 1 & 8 & 27\\ 1 & 1 & 16 & 81 \end{bmatrix}\begin{bmatrix} A\\ B\\ C\\ D \end{bmatrix}=\begin{bmatrix} 0\\ 0\\ 6\\ 120 \end{bmatrix}.$$

Hence,

$$\begin{aligned}
A&=\frac{7}{2}\\
B&=24\\
C&=-17\\
D&=\frac{9}{2}
\end{aligned}$$

and

$$-(A+B+C+D)=-15.$$

All of this gives

$$\begin{aligned}
f'''(x_0)=&\frac{Af(x_0-h)}{h^3}-\frac{(A+B+C+D)\,f(x_0)}{h^3}+\frac{Bf(x_0+h)}{h^3}\\
&+\frac{Cf(x_0+2h)}{h^3}+\frac{Df(x_0+3h)}{h^3}\\
=&\frac{7f(x_0-h)}{2h^3}-\frac{15f(x_0)}{h^3}+\frac{24f(x_0+h)}{h^3}-\frac{17f(x_0+2h)}{h^3}+\frac{9f(x_0+3h)}{2h^3}
\end{aligned}$$

with error

$$-5hf^{(4)}(x_0) + O(h^2).$$

Assuming a maximum roundoff error of μ in each function value, we have

$$\text{roundoff error} \leqslant \left(\frac{7}{2h^3} + \frac{15}{h^3} + \frac{24}{h^3} + \frac{17}{h^3} + \frac{9}{2h^3}\right)\mu = \frac{64\mu}{h^3}$$

Hence, the total error is

$$\Delta \lesssim 5h\left|f^{(4)}(x_0)\right| + \frac{64\mu}{h^3}$$

and so

$$\frac{d\Delta}{dh} = 0 \Rightarrow h \approx \left(\frac{192\mu}{5\left|f^{(4)}(x_0)\right|}\right)^{\frac{1}{4}}.$$

4. We have

$$\begin{aligned}
f(x_0-2h) &= f(x_0) - 2hf'(x_0) + \frac{4h^2}{2}f''(x_0) - \frac{8h^3}{6}f'''(x_0) + \frac{16h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)\\
f(x_0-h) &= f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)\\
f(x_0+h) &= f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f'''(x_0) + \frac{h^4}{24}f^{(4)}(x_0) + O\left(h^5\right)\\
f(x_0+2h) &= f(x_0) + 2hf'(x_0) + \frac{4h^2}{2}f''(x_0) + \frac{8h^3}{6}f'''(x_0) + \frac{16h^4}{24}f^{(4)}(x_0) + O\left(h^5\right).
\end{aligned}$$

Now

$$\begin{aligned}
&Af(x_0-2h) + Bf(x_0-h) + Cf(x_0+h) + Df(x_0+2h)\\
&= (A+B+C+D)f(x_0) + (-2A-B+C+2D)hf'(x_0)\\
&\quad + \left(2A + \frac{B}{2} + \frac{C}{2} + 2D\right)h^2f''(x_0) + \left(-\frac{4A}{3} - \frac{B}{6} + \frac{C}{6} + \frac{4D}{3}\right)h^3f'''(x_0)\\
&\quad + \left(\frac{16A}{24} + \frac{B}{24} + \frac{C}{24} + \frac{16D}{24}\right)h^4f^{(4)}(x_0) + O(h^5).
\end{aligned}$$

We require

$$\begin{aligned}
-2A - B + C + 2D &= 0\\
4A + B + C + 4D &= 2\\
-8A - B + C + 8D &= 0\\
16A + B + C + 16D &= 72,
\end{aligned}$$

which gives

$$\begin{bmatrix} -2 & -1 & 1 & 2 \\ 4 & 1 & 1 & 4 \\ -8 & -1 & 1 & 8 \\ 16 & 1 & 1 & 16 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 72 \end{bmatrix}.$$

Hence,

$$\begin{aligned} A &= \frac{35}{12} \\ B &= -\frac{32}{3} \\ C &= -\frac{32}{3} \\ D &= \frac{35}{12} \end{aligned}$$

and

$$-\left(A+B+C+D\right) = \frac{31}{2}.$$

All of this gives

$$\begin{aligned} f''\left(x_0\right) =& \frac{Af(x_0-2h)}{h^2} + \frac{Bf(x_0-h)}{h^2} - \frac{\left(A+B+C+D\right)f\left(x_0\right)}{h^2} \\ &+ \frac{Cf(x_0+h)}{h^2} + \frac{Df(x_0+2h)}{h^2} \\ =& \frac{35f(x_0-2h)}{12h^2} - \frac{32f(x_0-h)}{3h^2} + \frac{31f\left(x_0\right)}{2h^2} - \frac{32f(x_0+h)}{3h^2} + \frac{35f(x_0+2h)}{12h^2} \end{aligned}$$

with error

$$-3h^2 f^{(4)}\left(x_0\right) + O(h^3).$$

22 Boundary Value Problems

We consider the solution of two-point nonlinear boundary-value problems, and Poisson's equation in two dimensions.

22.1 Two-point Boundary Value Problem

We solve

$$f\left(x, y, y', y''\right) = 0 \tag{11}$$

on $[a, b]$, subject to

$$\begin{aligned} y\left(a\right) &= \alpha \\ y\left(b\right) &= \beta. \end{aligned}$$

We assume that $f\left(x, y, y', y''\right)$ is nonlinear.

22.1.1 Discretization and Finite Differences

We discretize $[a, b]$ into N subintervals, defined by the $N+1$ nodes $\{x_0, x_1, \ldots, x_N\}$. Clearly, $x_0 = a$ and $x_N = b$. At each of the $N-1$ interior nodes $\{x_1, \ldots, x_{N-1}\}$ we approximate the derivatives in (11) by finite differences, as in

$$\begin{aligned} y'\left(x_i\right) &\approx \frac{y\left(x_{i+1}\right) - y\left(x_{i-1}\right)}{2h} && \left[-\frac{h^2 y'''\left(\xi\right)}{6}\right] \\ y''\left(x_i\right) &\approx \frac{y\left(x_{i+1}\right) - 2y\left(x_i\right) + y\left(x_{i-1}\right)}{h^2} && \left[-\frac{h^2 y^{(4)}\left(\zeta\right)}{12}\right], \end{aligned}$$

where h is the node spacing, and $i = 1, \ldots, N-1$. The terms in brackets are the relevant approximation errors. Using the symbol w to denote the numerical solution, we find the system of $N-1$ nonlinear algebraic equations

$$f_i\left(x_i, w_{i-1}, w_i, w_{i+1}\right) \equiv f\left(x_i, w_i, \frac{w_{i+1} - w_{i-1}}{2h}, \frac{w_{i+1} - 2w_i + w_{i-1}}{h^2}\right) = 0,$$

with $w_0 = \alpha$ and $w_N = \beta$. These are then solved using a root-finding method, such as Newton's Method, with a strict tolerance.

22.1.2 Error Control

For small h, it is known that the method described above has absolute error $A_i h^2$ at each node (where A_i is independent of h). Let $w_i\left(h\right)$ denote the numerical solution at x_i with stepsize h. Now, let $w_j\left(\frac{h}{2}\right)$ denote the numerical

solution at x_j with stepsize $\frac{h}{2}$. Note that the set of nodes $\{x_j\}$ contains $2N+1$ elements. Each node x_j, j odd, lies between two adjacent nodes of the set $\{x_i\}$. The nodes x_j, j even, coincide with $\{x_i\}$. The difference between the numerical solutions at these coincident nodes gives

$$w_i(h) - w_j\left(\frac{h}{2}\right) = A_i\left(h^2 - \frac{h^2}{4}\right).$$

From this we can find $A_M \equiv \max|A_i|$, and then demand

$$A_M (h^*)^2 \leqslant \varepsilon,$$

where ε is a tolerance, which gives

$$h^* \leqslant \sqrt{\frac{\varepsilon}{A_M}}.$$

We insist that this new stepsize must subdivide $[a, b]$ into an integer number of subintervals, and so

$$h^* = \frac{b-a}{N^*}$$
$$N^* = \left\lceil \sqrt{\frac{A_M (b-a)^2}{\varepsilon}} \right\rceil.$$

22.2 Nonlinear Shooting Method

A well-known algorithm for finding a numerical solution to the nonlinear boundary-value problem

$$\begin{aligned} &y'' = f(x, y, y') \\ &x \in [a, b] \qquad y(a) = \alpha \qquad y(b) = \beta \end{aligned} \tag{12}$$

is the iterative nonlinear *shooting* method. This method requires the solution of an initial-value problem of the form

$$\begin{aligned} &y' = z \\ &z' = f(x, y, z) \\ &x \in [a, b] \qquad y(a) = \alpha \qquad z(a) = y'(a) = t. \end{aligned} \tag{13}$$

The shooting method for (12) is summarized by the iterative procedure

$$t_{k+1} = t_k - (w_k(b) - \beta)\left(\frac{t_k - t_{k-1}}{w_k(b) - w_{k-1}(b)}\right).$$

In this expression k indicates the iteration count, t_k is the approximate value of $y'(a)$ after iteration k, and $w_k(b)$ is the approximation to β obtained using an Runge-Kutta method applied to (13), with $z(a) = t_k$. Note that the term in parentheses on the RHS is a finite-difference approximation to the reciprocal of the derivative dy/dt. This procedure requires that two initial guesses for the slope, t_1 and t_2, are specified. The iteration then proceeds until the *residual* $|w_{k+1}(b) - \beta|$ is less than a user-specified tolerance ε. The resulting t_{k+1} is taken as the slope $y'(a)$, and the Runge-Kutta solution of (13), with $z(a) = t_{k+1}$, is taken as the solution to (12). The quality of the solution depends on the error control implemented for the Runge-Kutta method.

22.3 Comments

Often, a BVP has the form

$$y'' = g(x, y, y'). \tag{14}$$

In such a case, the FDM solves (14) with

$$y(a) = \alpha \quad y(b) = \beta,$$

whereas the SM solves (14) with

$$y(a) = \alpha \quad y(b) = w_{k+1}(b).$$

If $|w_{k+1}(b) - \beta| \ll 1$, then the two solutions could be regarded as compatible.

22.4 Poisson's Equation

Poisson's equation in two dimensions is an *elliptic* partial differential equation (PDE) of the form

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y).$$

The solution $u(x, y)$ is sought on a region D of the xy plane; we will consider here the case when D is rectangular. Boundary conditions are imposed in the form of u, u_x or u_y, or combinations of these, on the boundary of D.

22.4.1 Discretization and Finite Differences

For ease, we will assume that D is located in the positive quadrant of the xy plane, such that its lower left corner is at the origin. Now, define D by

$$D = \{(x, y) |\ 0 < x < L_1,\ 0 < y < L_2\}.$$

The horizontal dimension of D is discretized as

$$x_i = ih \quad i = 0, 1, \ldots, N,$$

where h is a suitable stepsize, and the vertical dimension of D is discretized as

$$y_j = jk \quad j = 0, 1, \ldots, M,$$

where k is a suitable stepsize. Note that the points (x_i, y_j) form a rectilinear grid of nodes on D. There are $(N+1)(M+1)$ nodes in total. The nodes (x_i, y_j) with $i = 1, 2, \ldots, N-1$ and $j = 1, 2, \ldots, M-1$ are *interior* to D; the remaining $2N + 2M$ nodes are on the boundary of D.

At each interior node, Poisson's equation is written using finite differences, as in

$$\frac{w_{i+1}^j - 2w_i^j + w_{i-1}^j}{h^2} + \frac{w_i^{j+1} - 2w_i^j + w_i^{j-1}}{k^2} = f(x_i, y_j), \tag{15}$$

where w_i^j denotes the numerical solution at (x_i, y_j), i.e. $w_i^j \approx u(x_i, y_j)$. This finite difference expression can be written for each of the $(N-1)(M-1)$ interior nodes, yielding $(N-1)(M-1)$ linear algebraic equations in the w_i^j. since the finite difference approximations to the derivatives have second-order errors, the error in the numerical solution will also be second-order, provided that h and k are sufficiently small.

A simple manipulation of (15) gives

$$-w_{i+1}^j - w_{i-1}^j + \left(2 + \frac{2h^2}{k^2}\right) w_i^j - \left(\frac{h^2}{k^2}\right) w_i^{j+1} - \left(\frac{h^2}{k^2}\right) w_i^{j-1} = -h^2 f(x_i, y_j)$$

and, if $h = k$, we have

$$-w_{i+1}^j - w_{i-1}^j + 4w_i^j - w_i^{j+1} - w_i^{j-1} = -h^2 f(x_i, y_j) \tag{16}$$

for $i = 1, 2, \ldots, N-1$ and $j = 1, 2, \ldots, M-1$.

22.4.2 Error Control

Let $w(x_i, y_j; h, k)$ denote the approximate solution at an interior node (x_i, y_j) obtained using (15). At each interior node it can be shown that, if we use

second-order derivative approximations and if h and k are small enough, we have

$$w\left(x_i, y_j; h, k\right) = u\left(x_i, y_j\right) + A_{ij}h^2 + B_{ij}k^2 + \ldots, \tag{17}$$

where A_{ij} and B_{ij} are independent of h and k, and are functions of x and y only. Hence, A_{ij} and B_{ij} can vary from node to node (as indicated by their subscripts).

Now consider the numerical solutions at the same node on two finer grids

$$w\left(x_i, y_j; \frac{h}{\theta_1}, \frac{k}{\eta_1}\right) = u\left(x_i, y_j\right) + A_{ij}\frac{h^2}{\theta_1^2} + B_{ij}\frac{k^2}{\eta_1^2} + \ldots$$
$$w\left(x_i, y_j; \frac{h}{\theta_2}, \frac{k}{\eta_2}\right) = u\left(x_i, y_j\right) + A_{ij}\frac{h^2}{\theta_2^2} + B_{ij}\frac{k^2}{\eta_2^2} + \ldots,$$

where $\{\theta_1, \theta_2, \eta_1, \eta_2\} \in \mathbb{Z}^+$, and

$$1 < \theta_1, \eta_1 < \theta_2, \eta_2.$$

These finer grids have smaller spacing between nodes. Obviously, $\{\theta_1, \theta_2, \eta_1, \eta_2\}$ are chosen so that the node (x_i, y_j) is common to all three grids. Neglecting higher-order terms, we find

$$w\left(x_i, y_j; h, k\right) - w\left(x_i, y_j; \frac{h}{\theta_1}, \frac{k}{\eta_1}\right) = A_{ij}\left(h^2 - \frac{h^2}{\theta_1^2}\right) + B_{ij}\left(k^2 - \frac{k^2}{\eta_1^2}\right)$$
$$w\left(x_i, y_j; h, k\right) - w\left(x_i, y_j; \frac{h}{\theta_2}, \frac{k}{\eta_2}\right) = A_{ij}\left(h^2 - \frac{h^2}{\theta_2^2}\right) + B_{ij}\left(k^2 - \frac{k^2}{\eta_2^2}\right),$$

which yields the linear system

$$\begin{bmatrix} h^2 - \frac{h^2}{\theta_1^2} & k^2 - \frac{k^2}{\eta_1^2} \\ h^2 - \frac{h^2}{\theta_2^2} & k^2 - \frac{k^2}{\eta_2^2} \end{bmatrix} \begin{bmatrix} A_{ij} \\ B_{ij} \end{bmatrix} = \begin{bmatrix} w\left(x_i, y_j; h, k\right) - w\left(x_i, y_j; \frac{h}{\theta_1}, \frac{k}{\eta_1}\right) \\ w\left(x_i, y_j; h, k\right) - w\left(x_i, y_j; \frac{h}{\theta_2}, \frac{k}{\eta_2}\right) \end{bmatrix}. \tag{18}$$

Invertibility of this system requires that the determinant of the coefficient matrix be nonzero, which gives

$$\frac{h^2k^2\left(\eta_1^2\theta_1^2\eta_2^2 + \theta_1^2\eta_2^2\theta_2^2 + \eta_1^2\theta_2^2 - \eta_1^2\theta_1^2\theta_2^2 - \eta_1^2\eta_2^2\theta_2^2 - \eta_2^2\theta_1^2\right)}{\left(\theta_1\theta_2\eta_1\eta_2\right)^2} \neq 0$$

so that

$$\eta_1^2\theta_1^2\eta_2^2 + \theta_1^2\eta_2^2\theta_2^2 + \eta_1^2\theta_2^2 - \eta_1^2\theta_1^2\theta_2^2 - \eta_1^2\eta_2^2\theta_2^2 - \eta_2^2\theta_1^2 \neq 0.$$

We can choose values for $\{\theta_1, \eta_1, \theta_2\}$ and then solve this quadratic equation to find the values of η_2 that are not permissible. For example, choosing

$$\{\theta_1, \eta_1, \theta_2\} = \{2, 2, 3\}$$

gives

$$\eta_2 \neq 3,$$

so that $\eta_2 = 4$, for example, would ensure invertibility.

Next, we define the node-dependent tolerance

$$\delta_{ij} \equiv \max\left\{\delta_A, \delta_R \left|u\left(x_i, y_j\right)\right|\right\} = \delta_R \max\left\{\frac{\delta_A}{\delta_R}, \left|u\left(x_i, y_j\right)\right|\right\}, \tag{19}$$

where δ_A and δ_R are user-defined. Clearly, δ_{ij} is equal to δ_A when $0 \leqslant \left|u\left(x_i, y_j\right)\right| < \frac{\delta_A}{\delta_R}$, and $\delta_R \left|u\left(x_i, y_j\right)\right|$ otherwise. Often, one chooses $\delta_A = \delta_R$. Of course, the exact solution u is not known, so $u\left(x_i, y_j\right)$ in (19) is replaced with $w\left(x_i, y_j; \frac{h}{\theta_2}, \frac{k}{\eta_2}\right)$, which is assumed to be the most accurate of our available approximate solutions.

We then compute

$$h_{ij} = \sqrt{\frac{\delta_{ij}}{2\left|A_{ij}\right|}} \qquad k_{ij} = \sqrt{\frac{\delta_{ij}}{2\left|B_{ij}\right|}} \tag{20}$$

for each interior node, and choose

$$h^* = \sigma \min\left\{h_{ij}\right\} \qquad k^* = \sigma \min\left\{k_{ij}\right\},$$

where $\sigma < 1$ is a safety factor, intended to cater for the omission of the higher-order terms. The need for the option of the absolute tolerance δ_A in (19) is clear from (20): without such an option, particularly if $u\left(x_i, y_j\right)$ is close to zero, $\delta_{ij} = \delta_R \left|u\left(x_i, y_j\right)\right|$ would be close to zero, leading to very small and impractical values for h_{ij} and k_{ij}. Furthermore, if $u\left(x_i, y_j\right)$ is identically zero, then the stepsizes h_{ij} and k_{ij} would also be zero, which is untenable. The absolute tolerance δ_A thus provides a lower limit on the discretization parameters, since we always have

$$h_{ij} \geqslant \sqrt{\frac{\delta_A}{2\left|A_{ij}\right|}} \qquad k_{ij} \geqslant \sqrt{\frac{\delta_A}{2\left|B_{ij}\right|}}.$$

The discretization parameters are refined as

$$h^* \leftarrow \frac{b-a}{\left\lceil \frac{b-a}{h^*} \right\rceil} \qquad k^* \leftarrow \frac{d-c}{\left\lceil \frac{d-c}{k^*} \right\rceil}.$$

These refinements ensure that $b-a$ and $d-c$ are integer multiples of h^* and k^*.

Finally, the approximate solutions

$$w\left(x_i, y_j; h^*, k^*\right)$$

on the new grid defined by h^* and k^* are found. With these discretization parameters, we have

$$|A_{ij}|\left(h^*\right)^2 + |B_{ij}|\left(k^*\right)^2 \leqslant \frac{\delta_{ij}}{2} + \frac{\delta_{ij}}{2} = \delta_{ij}.$$

23 Exercises

1. Solve
$$\nabla^2 u\,(x, y) = 4$$
on
$$R = \{(x, y) \mid 0 < x < 1,\ 0 < y < 1\}$$
using the discretization
$$x_i = \frac{i}{3} \qquad\qquad y_j = \frac{j}{3}$$
for $i, j = 0, 1, \ldots, 3$, subject to the boundary conditions
$$\begin{aligned} u\,(x, 0) &= x^2 \\ \left.\frac{\partial u}{\partial y}\right|_{(x,1)} &= 2\,(1 - x) \\ u\,(0, y) &= y^2 \\ \left.\frac{\partial u}{\partial x}\right|_{(1,y)} &= 2\,(1 - y)\,. \end{aligned}$$
Make use of second-order approximations to the derivatives in the PDE and the boundary conditions. Compare the approximate solution with the actual solution
$$u\,(x, y) = (x - y)^2$$
and comment. Use a single-index notation for the approximate solution, as in
$$w_k = w_i^j$$
where $k = 13 + i - 4j$.

2. Set up a linear system whose solution approximates the solution of
$$\nabla^2 u\,(x, y) = 2y$$
on
$$A = \{(x, y) \mid 0 < x < 1,\ 0 < y < 2\}$$

subject to the boundary conditions

$$\begin{aligned} u(x,0) &= x \\ \left.\frac{\partial u}{\partial y}\right|_{(x,2)} &= x^2 \\ u(1,y) &= 1+y \\ \left.\frac{\partial u}{\partial x}\right|_{(0,y)} &= 1. \end{aligned}$$

Use the discretization

$$x_i = \frac{i}{3} \qquad y_j = \frac{2j}{3}$$

for $i, j = 0, 1, \ldots, 3$. Make use of second-order approximations to the derivatives in the PDE and in the boundary conditions. Compare the approximate solution with the actual solution

$$u(x,y) = x^2 y + x$$

and comment. Use a single-index notation for the approximate solution, as in

$$w_k = w_i^j$$

where $k = 13 + i - 4j$.

3. Assume $h = k \ll 1$. Note that in (16) the multiplication by h^2 should yield an error term of $O(h^4)$, since the derivative approximations have error $O(h^2)$. However, it is known that the error, despite the multiplication by h^2, is $O(h^2)$. Explain.

24 Solutions

1. From the boundary conditions we have

$$w_{13} = 0 \quad w_{14} = \frac{1}{9} \quad w_{15} = \frac{4}{9} \quad w_{16} = 1$$
$$w_9 = \frac{1}{9} \quad w_5 = \frac{4}{9} \quad w_1 = 1$$

The discretization parameters are

$$h = \frac{1}{3} \qquad k = \frac{1}{3}$$

The discrete equation at the interior points is

$$2\left[\left(\frac{h}{k}\right)^2 + 1\right] w_i^j - \left(w_{i+1}^j + w_{i-1}^j\right) - \left(\frac{h}{k}\right)^2 \left(w_i^{j+1} + w_i^{j-1}\right) = -4h^2$$
$$\Rightarrow 4w_i^j - \left(w_{i+1}^j + w_{i-1}^j\right) - \left(w_i^{j+1} + w_i^{j-1}\right) = -\frac{4}{9} \tag{21}$$

where i is the "x-index" and j is the "y-index". This yields

$$4w_6 - w_5 - w_7 - w_2 - w_{10} = -\frac{4}{9}$$
$$4w_7 - w_3 - w_{11} - w_6 - w_8 = -\frac{4}{9}$$
$$4w_{10} - w_6 - w_{14} - w_9 - w_{11} = -\frac{4}{9}$$
$$4w_{11} - w_7 - w_{15} - w_{10} - w_{12} = -\frac{4}{9}$$

The derivative boundary condition at $y = 1$ is handled using the backward-difference formula

$$u_y(x_i, 1) \approx \frac{1}{2k}\left[u(x_i, y_1) - 4u(x_i, y_2) + 3u(x_i, y_3)\right] \qquad i = 1, 2 \tag{22}$$

which gives

$$w_{10} - 4w_6 + 3w_2 = \frac{8}{9}$$
$$w_{11} - 4w_7 + 3w_3 = \frac{4}{9}$$

Similarly, the derivative boundary condition at $x = 1$ is handled with

$$u_x(1, y_j) \approx \frac{1}{2h}\left[u(x_1, y_j) - 4u(x_2, y_j) + 3u(x_3, y_j)\right] \qquad j = 1, 2 \tag{23}$$

which gives

$$w_{10} - 4w_{11} + 3w_{12} = \frac{8}{9}$$
$$w_6 - 4w_7 + 3w_8 = \frac{4}{9}$$

These eight equations, together with the boundary values, yield the linear system

$$A\bar{w} = \bar{b}$$

where

$$A = \begin{bmatrix} -1 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 3 & 0 & -4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & -4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 & 3 \\ 0 & 0 & 1 & -4 & 3 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{w} = \begin{bmatrix} w_2 \\ w_3 \\ w_6 \\ w_7 \\ w_8 \\ w_{10} \\ w_{11} \\ w_{12} \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} 0 \\ -\frac{4}{9} \\ -\frac{2}{9} \\ 0 \\ \frac{8}{9} \\ \frac{4}{9} \\ \frac{8}{9} \\ \frac{4}{9} \end{bmatrix}.$$

This linear system gives

$$\begin{bmatrix} w_2 \\ w_3 \\ w_6 \\ w_7 \\ w_8 \\ w_{10} \\ w_{11} \\ w_{12} \end{bmatrix} = \begin{bmatrix} 0.4444 \\ 0.1111 \\ 0.1111 \\ 0.0000 \\ 0.1111 \\ 0.0000 \\ 0.1111 \\ 0.4444 \end{bmatrix} \tag{24}$$

(the condition number of A is $\sim$31). The use of numerical software to solve this system is acceptable. For w_4 we have

$$u_y(1, 1) \approx w_{12} - 4w_8 + 3w_4 = 0 \quad \text{or} \quad u_x(1, 1) \approx w_2 - 4w_3 + 3w_4 = 0$$

from the derivative boundary conditions. Both yield

$$w_4 = 0.$$

The difference equation (21) has an error of the form

$$\frac{h^4}{12} u_{xxxx}(\epsilon, \zeta) + \frac{h^2 k^2}{12} u_{yyyy}(\varepsilon, \xi)$$

where $\epsilon, \zeta, \varepsilon, \xi$ are appropriate constants. The errors in (22) and (23) have the form

$$\frac{k^2}{3} u_{yyy}(\eta, \lambda) \quad \text{and} \quad \frac{h^2}{3} u_{xxx}(\delta, \gamma),$$

respectively, where $\delta, \gamma, \eta, \lambda$ are appropriate constants. The exact solution is

$$u(x, y) = (x - y)^2.$$

The numerical solution (24) is seen to be exact. This is easily understood by appealing to the error expressions above: since $u(x, y)$ is a multinomial of degree two, 3rd- and 4th-order derivatives of u are zero.

2. From the boundary conditions we have

$$\begin{array}{cccc} w_{13} = 0 & w_{14} = \dfrac{1}{3} & w_{15} = \dfrac{2}{3} & w_{16} = 1 \\ w_4 = 3 & w_8 = \dfrac{7}{3} & w_{12} = \dfrac{5}{3}. & \end{array}$$

The discretization parameters are

$$h = \frac{1}{3} \qquad k = \frac{2}{3}.$$

The discrete equation at the interior points is

$$\begin{aligned} & 2\left[\left(\frac{h}{k}\right)^2 + 1\right] w_i^j - \left(w_{i+1}^j + w_{i-1}^j\right) - \left(\frac{h}{k}\right)^2 \left(w_i^{j+1} + w_i^{j-1}\right) = -2y_i h^2 \\ \Rightarrow\ & \frac{10}{4} w_i^j - \left(w_{i+1}^j + w_{i-1}^j\right) - \frac{1}{4}\left(w_i^{j+1} + w_i^{j-1}\right) = -\frac{2y_i}{9} \\ \Rightarrow\ & 10 w_i^j - 4\left(w_{i+1}^j + w_{i-1}^j\right) - \left(w_i^{j+1} + w_i^{j-1}\right) = -\frac{8y_i}{9}, \end{aligned}$$

where i is the "x-index" and j is the "y-index". This yields

$$\begin{aligned}
10w_6 - 4w_5 - 4w_7 - w_2 - w_{10} &= -\frac{32}{27} \\
\Rightarrow 10w_6 - 4w_5 - 4w_7 - w_2 - w_{10} &= -\frac{32}{27} \\
10w_7 - 4w_6 - 4w_8 - w_3 - w_{11} &= -\frac{32}{27} \\
\Rightarrow 10w_7 - 4w_6 - w_{11} - w_3 &= -\frac{32}{27} + \frac{28}{3} = \frac{220}{27} \\
10w_{10} - 4w_9 - 4w_{11} - w_6 - w_{14} &= -\frac{16}{27} \\
\Rightarrow 10w_{10} - 4w_9 - 4w_{11} - w_6 &= -\frac{16}{27} + \frac{4}{3} = \frac{20}{27} \\
10w_{11} - 4w_{10} - 4w_{12} - w_7 - w_{15} &= -\frac{16}{27} \\
\Rightarrow 10w_{11} - 4w_{10} - w_7 &= -\frac{16}{27} + \frac{20}{3} + \frac{2}{3} = \frac{182}{27}.
\end{aligned}$$

The derivative boundary condition at $y = 2$ is handled using the backward-difference formula

$$u_y(x_i, 2) \approx \frac{1}{2k}\left[u\left(x_i, \frac{2}{3}\right) - 4u\left(x_i, \frac{4}{3}\right) + 3u(x_i, 2)\right] = x_i^2,$$

which gives

$$\begin{aligned}
w_{10} - 4w_6 + 3w_2 &= \frac{4}{27} \\
w_{11} - 4w_7 + 3w_3 &= \frac{16}{27}.
\end{aligned}$$

The derivative boundary condition at $x = 0$ is handled using the forward-difference formula

$$u_x(0, y_i) \approx \frac{1}{2h}\left[-3u(0, y_i) + 4u\left(\frac{1}{3}, y_i\right) - u\left(\frac{2}{3}, y_i\right)\right] = 1,$$

which gives

$$\begin{aligned}
-3w_9 + 4w_{10} - w_{11} &= \frac{2}{3} \\
-3w_5 + 4w_6 - w_7 &= \frac{2}{3}.
\end{aligned}$$

For w_1 we have

$$u_y(0,2) \approx w_9 - 4w_5 + 3w_1 = 0 \quad \text{or} \quad u_x(0,2) \approx -3w_1 + 4w_2 - w_3 = \frac{2}{3}$$

from the derivative boundary conditions. Either of these, together with the previous eight linear equations obtained, yields a 9×9 linear system which can be solved for $\{w_1, w_2, w_3, w_5, w_6, w_7, w_9, w_{10}, w_{11}\}$. The error analysis in exercise #1 also holds here; since $u(x,y) = x^2y + x$ is a multinomial of degree two, 3rd- and 4th-order derivatives of u are zero,and so the numerical solution is exact.

3. From (16), with the true solution u in place of the numerical solution w, and the errors in the derivative approximations explicitly shown (neglecting higher-order terms), we have

$$\begin{aligned} &u(x_{i+1}, y_j) - 4u(x_i, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1}) \\ &= h^2 f(x_i, y_j) + \frac{h^4}{12}\left[\left.\frac{\partial^4 u}{\partial x^4}\right|_{ij} + \left.\frac{\partial^4 u}{\partial y^4}\right|_{ij}\right] \end{aligned}$$

at each interior node. These equations can be combined into a linear system of the form

$$Lu = F + E \tag{25}$$

where L is a coefficient matrix, F is a vector containing $h^2 f$ and any relevant boundary values, and the vector E is an error term whose entries are $O(h^4)$. Neglecting the error term, the system becomes

$$Lw = F$$

where w denotes the solution to this particular system, which is taken as the approximate solution to the original PDE. Clearly, w is the solution obtained when (25) is solved in the absence of E. These two systems give

$$w - u = L^{-1}E$$

and, naïvely, one might expect that this error is $O(h^4)$. However, it can be shown that the spectral radius of L^{-1} is

$$\rho\left(L^{-1}\right) = \frac{1}{8\sin^2\left(\frac{\pi h}{2}\right)},$$

and so

$$\begin{aligned}\left\|L^{-1}E\right\| &\leqslant \left\|L^{-1}\right\| \|E\| \\ &= \left|\rho\left(L^{-1}\right)\right| \left|O\left(h^4\right)\right| \\ &= \frac{\left|O\left(h^4\right)\right|}{8\sin^2\left(\frac{\pi h}{2}\right)} \\ &\simeq \frac{\left|O\left(h^4\right)\right|}{2\pi^2 h^2} \\ &= O\left(h^2\right)\end{aligned}$$

if h is small enough. Hence, the approximation error in w is $O\left(h^2\right)$. The norm used here is the Euclidean norm. This result does suggest a way of deciding on values of h and k for use in the error control algorithm: choose $h = k$ such that $\sin^2\left(\frac{\pi h}{2}\right)$ is very close to $\pi^2 h^2/4$. Then the assumption that the error is $O\left(h^2\right)$ is valid, and the error control algorithm presented above will be reliable.

25 Parabolic PDE

The parabolic PDE that we consider here is the one-dimensional diffusion equation

$$\begin{gathered} \frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} \\ 0 < x < L,\ t > 0 \end{gathered} \tag{26}$$

where $u = u(x,t)$, subject to the conditions

$$\begin{gathered} u(0,t) = \alpha,\ u(L,t) = \beta \quad \text{for} \quad t > 0 \\ u(x,0) = f(x) \quad \text{for} \quad 0 \leqslant x \leqslant L. \end{gathered}$$

We will approximate the derivatives in (26) with finite differences, and find a numerical solution on a discrete grid. Note that the first of the conditions above are boundary values, and the second condition is an initial value. Hence, (26) actually constitutes a mixed "BVP-IVP" problem, so to speak.

25.1 Discretization and Finite Differences

It is clear that we seek a solution on a region of the xt plane. We discretize the interval $[0, L]$ in the usual way, i.e.

$$x_i = ih \quad i = 0, 1, \ldots, N,$$

where h is a suitable stepsize, and $Nh = L$. At the interior nodes ($i = 1, 2, \ldots, N-1$) we approximate the second derivative in (26) with the usual finite-difference expression.

Now, let

$$t_j = jk,$$

where k is a stepsize on the t-axis. At the node (x_i, t_j), approximate the time derivative in (26) by means of the finite-difference expression

$$\left.\frac{\partial u}{\partial t}\right|_{i,j} \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k}.$$

This allows us to write (26) in the discrete form, in terms of the numerical solution w,

$$\frac{w_i^{j+1} - w_i^j}{k} = a^2 \left(\frac{w_{i+1}^j - 2w_i^j + w_{i-1}^j}{h^2} \right),$$

where $w_i^j \approx u(x_i, t_j)$, $w_{i+1}^j \approx u(x_{i+1}, t_j)$ etc. This equation can be rearranged to give

$$\begin{aligned} w_i^{j+1} &= w_i^j + \lambda\left(w_{i+1}^j - 2w_i^j + w_{i-1}^j\right) \\ &= \lambda w_{i+1}^j + (1-2\lambda)\, w_i^j + \lambda w_{i-1}^j, \end{aligned} \tag{27}$$

where $\lambda \equiv ka^2/h^2$. An equation like this can be written for each $i = 1, 2, \ldots, N-1$, yielding $N-1$ coupled algebraic equations, which can be written as

$$\begin{bmatrix} w_1^{j+1} \\ w_2^{j+1} \\ \vdots \\ \vdots \\ w_{N-1}^{j+1} \end{bmatrix} = \begin{bmatrix} 1-2\lambda & \lambda & 0 & \cdots & 0 \\ \lambda & 1-2\lambda & \lambda & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & \lambda & 1-2\lambda & \lambda \\ 0 & \cdots & 0 & \lambda & 1-2\lambda \end{bmatrix} \begin{bmatrix} w_1^j \\ w_2^j \\ \vdots \\ \vdots \\ w_{N-1}^j \end{bmatrix} + \lambda \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}$$

or, more compactly,

$$\mathbf{w}^{j+1} = A(\lambda)\, \mathbf{w}^j + \lambda \mathbf{b}. \tag{28}$$

Note that the initial values are

$$\mathbf{w}^0 = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \end{bmatrix}$$

and the boundary values are, for each j,

$$w_0 = \alpha, \; w_N = \beta.$$

It is clear that (28) can be used to generate a sequence of solutions $\mathbf{w}^j$, for $j = 0, 1, \ldots$, which approximate the solution $u(x_i, t_j)$ of the original PDE. This method is known as the *Forward Difference Method.*

25.2 Truncation Error in the Forward Difference Method

We define the *local error* ε_i^{j+1} and the *local truncation error* τ_i^{j+1} by

$$k\tau_i^{j+1} = u_i^{j+1} - \left[\lambda u_{i+1}^j + (1-2\lambda)\, u_i^j + \lambda u_{i-1}^j\right] \equiv -\varepsilon_i^{j+1},$$

where $u_i^{j+1} \equiv u\left(x_i, t_{j+1}\right)$. Using the Taylor expansions

$$\begin{aligned}
u_i^{j+1} &= u_i^j + k\left(u_t\right)_i^j + \frac{k^2}{2}\left(u_{tt}\right)_i^j + \ldots \\
u_{i+1}^j &= u_i^j + h\left(u_x\right)_i^j + \frac{h^2}{2}\left(u_{xx}\right)_i^j + \frac{h^3}{6}\left(u_{xxx}\right)_i^j + \frac{h^4}{24}\left(u_{xxxx}\right)_i^j + \ldots \\
u_{i-1}^j &= u_i^j - h\left(u_x\right)_i^j + \frac{h^2}{2}\left(u_{xx}\right)_i^j - \frac{h^3}{6}\left(u_{xxx}\right)_i^j + \frac{h^4}{24}\left(u_{xxxx}\right)_i^j + \ldots
\end{aligned}$$

we find

$$\begin{aligned}
k\tau_i^{j+1} &= k\left(u_t\right)_i^j - h^2\lambda\left(u_{xx}\right)_i^j + \frac{k^2}{2}\left(u_{tt}\right)_i^j - \frac{h^4}{12}\lambda\left(u_{xxxx}\right)_i^j + \ldots \\
&= k\left(\left(u_t\right)_i^j - a^2\left(u_{xx}\right)_i^j\right) + \frac{k^2}{2}\left(u_{tt}\right)_i^j - \frac{a^2kh^2}{12}\left(u_{xxxx}\right)_i^j + \ldots .
\end{aligned}$$

From the PDE

$$u_t = a^2 u_{xx}$$

and so

$$\begin{aligned}
k\tau_i^{j+1} &= \frac{k^2}{2}\left(u_{tt}\right)_i^j - \frac{a^2kh^2}{12}\left(u_{xxxx}\right)_i^j + \ldots \\
\Rightarrow\ \tau_i^{j+1} &= \frac{k}{2}\left(u_{tt}\right)_i^j - \frac{a^2h^2}{12}\left(u_{xxxx}\right)_i^j + \ldots \\
&= O\left(k + h^2\right).
\end{aligned}$$

25.3 Stability of the Forward Difference Method

It can be shown, in a result reminiscent of that obtained for Runge-Kutta methods, that

$$\mathbf{w}^j - u\left(\mathbf{x}, t_j\right) = \boldsymbol{\varepsilon}^j + A\boldsymbol{\varepsilon}^{j-1} + \ldots + A^{j-2}\boldsymbol{\varepsilon}^2 + A^{j-1}\boldsymbol{\varepsilon}^1,$$

where $\mathbf{x} \equiv \left(x_1, \ldots, x_{N-1}\right)$, from which we deduce the condition

$$\|A\| < 1 \Rightarrow \rho\left(A\right) < 1$$

to prevent the amplification of local errors.

It is known that the eigenvalues of A are given by

$$\phi_i = 1 - 4\lambda\sin^2\left(\frac{i\pi}{2N}\right) \qquad i = 1, 2, \ldots, N-1$$

so that the stability condition becomes

$$-1 < 1 - 4\lambda \sin^2\left(\frac{i\pi}{2N}\right) < 1$$

for all $i = 1, 2, \ldots, N-1$. This gives

$$0 < \lambda \sin^2\left(\frac{i\pi}{2N}\right) < \frac{1}{2},$$

which is satisfied if $\lambda < 1/2$. This, in turn, gives

$$k < \frac{h^2}{2a^2},$$

thus providing a relationship between the stepsizes h and k that ensures stability.

25.4 The Backward Difference Method

The *Backward Difference Method* for the numerical solution of (26) is given by

$$-\lambda w_{i+1}^{j+1} + (1+2\lambda)\, w_i^{j+1} - \lambda w_{i-1}^{j+1} = w_i^j$$

or, in matrix form,

$$\mathbf{w}^{j+1} = B(\lambda)^{-1}\, \mathbf{w}^j + B(\lambda)^{-1}\, \lambda \mathbf{b}$$

where

$$B(\lambda) = \begin{bmatrix} 1+2\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & -\lambda & 1+2\lambda & -\lambda \\ 0 & \cdots & 0 & -\lambda & 1+2\lambda \end{bmatrix}.$$

It can be shown that this method has

$$\tau_i^{j+1} = O\left(k + h^2\right)$$

and is unconditionally stable, i.e. there is no restriction on k as far as stability is concerned.

25.5 The Crank-Nicolson Method

The *Crank-Nicolson Method* for (26) is the average of the Forward and Backward Difference Methods, and is given by

$$-\frac{\lambda}{2}w_{i+1}^{j+1}+(1+\lambda)\,w_i^{j+1}-\frac{\lambda}{2}w_{i-1}^{j+1}=\frac{\lambda}{2}w_{i+1}^{j}+(1-\lambda)\,w_i^{j}+\frac{\lambda}{2}w_{i-1}^{j}$$

$$\mathbf{w}^{j+1}=B\left(\frac{\lambda}{2}\right)^{-1}A\left(\frac{\lambda}{2}\right)\mathbf{w}^{j}+B\left(\frac{\lambda}{2}\right)^{-1}A\left(\frac{\lambda}{2}\right)\lambda\mathbf{b}.$$

It has truncation error $O\left(k^2+h^2\right)$ and is unconditionally stable.

25.6 A Comment on Local Truncation Error

Strictly speaking, the local truncation error for the above methods is defined as

$$\tau_i^{j+1}\equiv\left[\left(\frac{w_i^{j+1}-w_i^{j}}{k}\right)-G\left(w\right)\right]_{w\to u},$$

where $G\left(w\right)$ is some function uniquely associated with the method under consideration, and the approximate values are replaced with exact values. This gives

$$k\tau_i^{j+1}=u_i^{j+1}-u_i^{j}-kG\left(u\right),$$

where $u_i^{j+1}\equiv u\left(x_i,t_{j+1}\right)$ etc. For the Forward Difference Method we have

$$kG\left(u\right)=\lambda u_{i+1}^{j}-2\lambda u_i^{j}+\lambda u_{i-1}^{j},$$

for the Backward Difference Method we have

$$kG\left(u\right)=\lambda u_{i+1}^{j+1}-2\lambda u_i^{j+1}+\lambda u_{i-1}^{j+1},$$

and for the Crank-Nicolson Method we have

$$kG\left(u\right)=\frac{\lambda}{2}u_{i+1}^{j+1}-\lambda u_i^{j+1}+\frac{\lambda}{2}u_{i-1}^{j+1}+\frac{\lambda}{2}u_{i+1}^{j}-\lambda u_i^{j}+\frac{\lambda}{2}u_{i-1}^{j}.$$

Now, consider the Forward Difference Method in the form

$$\begin{aligned}w_i^{j+1}&=w_i^{j}+\lambda\left(w_{i+1}^{j}-2w_i^{j}+w_{i-1}^{j}\right)\\&=w_i^{j}+kG\left(w\right).\end{aligned}$$

This is reminiscent of a one-step method. The local error ε_i^{j+1} for such a method is

$$\varepsilon_i^{j+1}\equiv\left[u_i^{j}+kG\left(u\right)\right]-u_i^{j+1}=-k\tau_i^{j+1}.$$

Similar results obtain for the Backward Difference and Crank-Nicolson Methods.

Note that, from the structure of (26), $G(w)$ represents a "method-specific" discrete approximation to $a^2 u_{xx}$. For the Forward Difference Method this is obvious:

$$G(w) = \frac{a^2\left(w_{i+1}^j - 2w_i^j + w_{i-1}^j\right)}{h^2}.$$

For the Backward Difference and Crank-Nicolson Methods we have

$$G(w) = \frac{a^2\left(w_{i+1}^{j+1} - 2w_i^{j+1} + w_{i-1}^{j+1}\right)}{h^2}$$

$$G(w) = \frac{a^2\left(w_{i+1}^{j+1} - 2w_i^{j+1} + w_{i-1}^{j+1} + w_{i+1}^j - 2w_i^j + \lambda w_{i-1}^j\right)}{2h^2},$$

where the implicit nature of these methods is clear.

26 Exercises

1. Derive the expression

$$\mathbf{w}^j - u\left(\mathbf{x}, t_j\right) = \boldsymbol{\varepsilon}^j + A\boldsymbol{\varepsilon}^{j-1} + \ldots + A^{j-2}\boldsymbol{\varepsilon}^2 + A^{j-1}\boldsymbol{\varepsilon}^1$$

   for the Forward Difference Method.

2. Use the *von Neumann ansatz* to investigate the stability of the Forward and Backward Difference Methods.

3. Investigate the stability of the Crank-Nicolson Method, using the eigenvalues of its coefficient matrices.

4. Determine the local truncation error for the Backward Difference Method.

5. Discretize (26) in a manner suitable for a Runge-Kutta method. Investigate the stability of this approach if an explicit Runge-Kutta method is used.

27 Solutions

1. The local error is

$$\boldsymbol{\varepsilon}^{j+1} \equiv \left[Au\left(\mathbf{x}, t_j\right) + \lambda \mathbf{b}\right] - u\left(\mathbf{x}, t_{j+1}\right).$$

and the global error is

$$\Delta^j \equiv \mathbf{w}^j - u\left(\mathbf{x}, t_j\right).$$

We will use the notation $u^j = u\left(\mathbf{x}, t_j\right)$ from this point onwards. Now,

$$\begin{aligned} \mathbf{w}^1 &= Au^0 + \lambda \mathbf{b} \\ \Rightarrow u^1 + \Delta^1 &= Au^0 + \lambda \mathbf{b} \\ \Rightarrow \Delta^1 &= \left[Au^0 + \lambda \mathbf{b}\right] - u^1 = \boldsymbol{\varepsilon}^1, \end{aligned}$$

$$\begin{aligned} \mathbf{w}^2 &= A\mathbf{w}^1 + \lambda \mathbf{b} \\ \Rightarrow u^2 + \Delta^2 &= Au^1 + A\Delta^1 + \lambda \mathbf{b} \\ \Rightarrow \Delta^2 &= \left[Au^1 + \lambda \mathbf{b}\right] - u^2 + A\Delta^1 = \boldsymbol{\varepsilon}^2 + A\boldsymbol{\varepsilon}^1, \end{aligned}$$

$$\begin{aligned} \mathbf{w}^3 &= A\mathbf{w}^2 + \lambda \mathbf{b} \\ \Rightarrow u^3 + \Delta^3 &= Au^2 + A\Delta^2 + \lambda \mathbf{b} \\ \Rightarrow \Delta^3 &= \left[Au^2 + \lambda \mathbf{b}\right] - u^3 + A\left(\boldsymbol{\varepsilon}^2 + A\boldsymbol{\varepsilon}^1\right) = \boldsymbol{\varepsilon}^3 + A\boldsymbol{\varepsilon}^2 + A^2\boldsymbol{\varepsilon}^1 \end{aligned}$$

and so on. Indeed, we have

$$\begin{aligned} \Delta^j &= \sum_{k=1}^{j} A^{j-k}\boldsymbol{\varepsilon}^k \\ &= \boldsymbol{\varepsilon}^j + A\boldsymbol{\varepsilon}^{j-1} + \ldots + A^{j-1}\boldsymbol{\varepsilon}^1. \end{aligned}$$

2. The *von Neumann ansatz is*

$$w_i^j = \xi^j e^{\mathbf{i}\theta i}$$

where i denotes the space index, j denotes the time index, and θ is a phase angle related to the grid spacing. We substitute this expression for w_i^j into the relevant difference equation and then determine the condition for which $|\xi| \leqslant 1$.

Forward Difference Method:

$$w_i^{j+1} = \left(1 - \frac{2a^2k}{h^2}\right) w_i^j + \frac{a^2k}{h^2}\left(w_{i+1}^j + w_{i-1}^j\right)$$

Using the von Neumann ansatz

$$w_i^j = \xi^j e^{\mathbf{i}\theta i}$$

with

$$\lambda \equiv \frac{a^2k}{h^2}$$

gives

$$\begin{aligned}
\xi\xi^j e^{\mathbf{i}\theta i} &= (1-2\lambda)\,\xi^j e^{\mathbf{i}\theta i} + \lambda\left(\xi^j e^{\mathbf{i}\theta i}e^{\mathbf{i}\theta} + \xi^j e^{\mathbf{i}\theta i}e^{-\mathbf{i}\theta}\right) \\
\Rightarrow\ \xi &= (1-2\lambda) + \lambda\left(e^{\mathbf{i}\theta} + e^{-\mathbf{i}\theta}\right) \\
&= (1-2\lambda) + \lambda\,(2\cos\theta) \\
&= 1 + 2\lambda\,(\cos\theta - 1)\,.
\end{aligned}$$

Now,

$$\begin{aligned}
|\xi| \leqslant 1 \ &\Rightarrow\ -1 \leqslant 1 + 2\lambda\,(\cos\theta - 1) \leqslant 1 \\
&\Rightarrow\ 0 \leqslant 2 + 2\lambda\,(\cos\theta - 1) \leqslant 2
\end{aligned}$$

The RHS of this inequality is always true. The LHS gives

$$-1 \leqslant \lambda\,(\cos\theta - 1) \ \Rightarrow\ \lambda\,(1 - \cos\theta) \leqslant 1.$$

Since the maximum value of $(1 - \cos\theta)$ is two, this inequality is always satisfied if

$$\lambda \leqslant \frac{1}{2}$$

Backward Difference Method:

$$(1+2\lambda)\,w_i^{j+1} - \lambda w_{i+1}^{j+1} - \lambda w_{i-1}^{j+1} = w_i^j$$

The von Neumann ansatz gives

$$\begin{aligned}
&(1+2\lambda)\,\xi^{j+1}e^{\mathbf{i}\theta i} - \lambda\left(\xi^{j+1}e^{\mathbf{i}\theta i}e^{\mathbf{i}\theta} + \xi^{j+1}e^{\mathbf{i}\theta i}e^{-\mathbf{i}\theta}\right) = \frac{\xi^{j+1}e^{\mathbf{i}\theta i}}{\xi} \\
\Rightarrow\ &\frac{1}{\xi} = 1 + 2\lambda\,(1 - \cos\theta) \\
\Rightarrow\ &|\xi| = \left|\frac{1}{1 + 2\lambda\,(1-\cos\theta)}\right|
\end{aligned}$$

and, since $(1 - \cos\theta) \geqslant 0$ and $\lambda > 0$, we have $|\xi| \leqslant 1$ always.

3. The eigenvalues of $A\left(\frac{\lambda}{2}\right)$ are

$$\phi_i = 1 - 2\lambda \sin^2\left(\frac{i\pi}{2N}\right) \qquad i = 1, 2, \ldots, N-1$$

and the eigenvalues of $B\left(\frac{\lambda}{2}\right)$ are

$$\varphi_i = 1 + 2\lambda \sin^2\left(\frac{i\pi}{2N}\right) \qquad i = 1, 2, \ldots, N-1$$

so that the eigenvalues of $B^{-1}\left(\frac{\lambda}{2}\right) A\left(\frac{\lambda}{2}\right)$ are

$$\vartheta_i = \frac{1 - 2\lambda \sin^2\left(\frac{i\pi}{2N}\right)}{1 + 2\lambda \sin^2\left(\frac{i\pi}{2N}\right)} \qquad i = 1, 2, \ldots, N-1.$$

Since $2\lambda \sin^2\left(\frac{i\pi}{2N}\right) > 0$ and $0 < \sin^2\left(\frac{i\pi}{2N}\right) < 1$, we have that

$$\left|1 - 2\lambda \sin^2\left(\frac{i\pi}{2N}\right)\right| < \left|1 + 2\lambda \sin^2\left(\frac{i\pi}{2N}\right)\right|$$

so that

$$|\vartheta_i| < 1$$

for all λ. This implies unconditional stability for the Crank-Nicolson Method.

4. Backward Difference Method local truncation error:

$$k\tau_i^{j+1} \equiv u_i^{j+1} - u_i^j - \left[\lambda u_{i+1}^{j+1} - 2\lambda u_i^{j+1} + \lambda u_{i-1}^{j+1}\right]$$

Expanding about (x_i, t_{j+1})

$$\begin{aligned}
u_i^j &= u_i^{j+1} - k\left(u_t\right)_i^{j+1} + \frac{k^2}{2}\left(u_{tt}\right)_i^{j+1} + \ldots \\
u_{i+1}^{j+1} &= u_i^{j+1} + h\left(u_x\right)_i^{j+1} + \frac{h^2}{2}\left(u_{xx}\right)_i^{j+1} + \frac{h^3}{6}\left(u_{xxx}\right)_i^{j+1} + \frac{h^4}{24}\left(u_{xxxx}\right)_i^{j+1} + \ldots \\
u_{i-1}^{j+1} &= u_i^{j+1} - h\left(u_x\right)_i^{j+1} + \frac{h^2}{2}\left(u_{xx}\right)_i^{j+1} - \frac{h^3}{6}\left(u_{xxx}\right)_i^{j+1} + \frac{h^4}{24}\left(u_{xxxx}\right)_i^{j+1} + \ldots,
\end{aligned}$$

we find

$$\begin{aligned}
k\tau_i^{j+1} &= k\left(u_t\right)_i^{j+1} - \frac{k^2}{2}\left(u_{tt}\right)_i^{j+1} - ka^2\left(u_{xx}\right)_i^{j+1} - \frac{kh^2a^2}{12}\left(u_{xxxx}\right)_i^{j+1} + \ldots \\
&= k\left(\left(u_t\right)_i^{j+1} - a^2\left(u_{xx}\right)_i^{j+1}\right) - \frac{k^2}{2}\left(u_{tt}\right)_i^{j+1} - \frac{kh^2a^2}{12}\left(u_{xxxx}\right)_i^{j+1} + \ldots.
\end{aligned}$$

From the PDE

$$u_t = a^2 u_{xx} \Rightarrow u_t - a^2 u_{xx} = 0,$$

and so

$$\begin{aligned} k\tau_i^{j+1} &= -\frac{k^2}{2}\left(u_{tt}\right)_i^{j+1} - \frac{kh^2a^2}{12}\left(u_{xxxx}\right)_i^{j+1} + \dots \\ \Rightarrow \tau_i^{j+1} &= -\frac{k}{2}\left(u_{tt}\right)_i^{j+1} - \frac{h^2a^2}{12}\left(u_{xxxx}\right)_i^{j+1} + \dots \end{aligned}$$

5. Discretization of the RHS of

$$u_t = a^2 u_{xx}$$

gives

$$\frac{dw_i}{dt} = \mu\left(w_{i-1} - 2w_i + w_{i+1}\right) \qquad i = 1, 2, ..., N-1$$

at the equidistant nodes $\{x_1, x_2, ..., x_{N-1}\}$. Here, the partial derivative with respect to t has been replaced with a total derivative. Also,

$$\mu \equiv \frac{a^2}{h^2}.$$

The boundary values give $w_0 = \alpha, w_N = \beta$, and so we have

$$\begin{aligned} \frac{dw_1}{dt} &= \mu\left(\alpha - 2w_1 + w_2\right) \equiv g_1 \\ \frac{dw_i}{dt} &= \mu\left(w_{i-1} - 2w_i + w_{i+1}\right) \equiv g_i \qquad i = 2, \dots, N-2 \\ \frac{dw_{N-1}}{dt} &= \mu\left(w_{N-2} - 2w_{N-1} + \beta\right) \equiv g_{N-1} \end{aligned}$$

The initial values are simply

$$w_i\left(t = 0\right) = f\left(x_i\right).$$

The above system of ODEs yields the Jacobian

$$J = \begin{bmatrix} \frac{\partial g_1}{\partial w_1} & \cdots & \frac{\partial g_1}{\partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial w_1} & \cdots & \frac{\partial g_n}{\partial w_n} \end{bmatrix} = \mu \begin{bmatrix} -2 & 1 & 0 & & & & \\ 1 & -2 & 1 & 0 & & & \\ 0 & 1 & -2 & 1 & 0 & & \\ & & & \ddots & & & \\ & & 0 & 1 & -2 & 1 & 0 \\ & & & 0 & 1 & -2 & 1 \\ & & & & 0 & 1 & -2 \end{bmatrix}.$$

The eigenvalues of J are

$$\phi_i = -2\mu + 2\mu \cos\left(\frac{i\pi}{N}\right) \qquad i = 1, 2, ..., N-1$$

and clearly

$$\max |\phi_i| \leqslant 4\mu.$$

Note that, since $-1 < \cos\left(\frac{i\pi}{N}\right) < 1$, ϕ_i is always negative. This means that ϕ_i is a stiff eigenvalue. Hence, for stability we require

$$k \leqslant \frac{\theta}{\max |\phi_i|},$$

where k is the Runge-Kutta stepsize, and where θ is the "real stiffness limit" for the *explicit* Runge-Kutta method used to solve the ODE system. For example, for Euler's method $\theta = 2$, and so

$$k \leqslant \frac{2}{4\mu} \Rightarrow \frac{ka^2}{h^2} \leqslant \frac{1}{2}.$$