SCI 4010 COMPUTATIONAL METHODS

MSc Course on Scientific Computing
Hong Kong Baptist University

©1999
001

Contents

3 Discretization **37**

3.1 Numerical Intergration 37
3.1.1 Trapezoidal Rule 37
3.1.2 Simpson's Rule 39
3.1.3 Romberg Integration 41
3.1.4 Adaptive Quadrature Method 42
3.2 Initial Value Problems 45
3.2.1 Euler's Method 47
3.2.2 Runge-Kutta Methods 50
3.2.3 Multistep Methods 53
3.2.4 Numerical Solutions of Systems of ODEs 56
3.2.5 Stability 56
3.3 Boundary Value Problems 62
3.3.1 Finite Difference Approximation 62
3.3.2 Shooting Method 67
3.4 Partial Differential Equation 70
3.4.1 Space and Time 70
3.4.2 Two Dimensional Problems 74

©1999
001

Chapter 3

Discretization

3.1 Numerical Intergration

By a "continuous problem", we mean one that uses data on a whole interval or, more generally, some region of higher dimensional space. In scientific computing, most of the problems are continuous. In order to solve a continuous problem on a computer, we need to approximate the problem in some way by a problem of "discrete type". That means only finite number of data is involved for input and output.

3.1.1 Trapezoidal Rule

Quadrature Rule

Among continuous problem perhaps the simplest example is the problem of finding the intergral

$$I(f) = \int_a^b f(x)dx\,. \tag{3.1}$$

The antiderivative $F(x)$ of the integrand $f(x)$ often has no explicit form, so that the formula

$$\int_a^b f(x)dx = F(b) - F(a)$$

for evaluating a definite integral is mainly of theoretical significance. Even if an explicit expression such as an infinite series were given, but this expression might be too complicated to evaluate. Moreover, in many cases, only a table of values $\{x_i, f(x_i)\}$ is given. Hence, we need to develop approximation of the form

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} \alpha_i f(x_i)$$

to the integral (3.1). One way to obtain such approximations for integrals, by a linear combination of function values of integrand at a finite number of points x_i, is called a `Quadrature`

`Rule`. Generally speaking, it is to replace the function $f(x)$ by some other functions whose integral is relatively easy to evaluate. The most common choices are polynomials or piecewise polynomials.

Trapezoidal Rule

The rule is based on approximating $f(x)$ by the straight line joining $[a, f(a)]$, $[b, f(b)]$. By integrating the formula for this straight line $y = p_1(x)$, we obtain the approximation.

$$\begin{aligned} I(f) &= \int_a^b f(x)dx \approx \int_a^b p_1(x)dx \\ &= \int_a^b \left(\frac{b-x}{b-a} f(x) + \frac{x-a}{b-a} f(b) \right) dx \\ &= \frac{b-a}{2}[f(a) + f(b)] = T(f) \end{aligned}$$

Error Term

The remainder of interpolation is

$$R(x) = f(x) - p_1(x) = \frac{f''(\xi_x)}{2}(x-a)(x-b), \qquad a < \xi_x < b$$

The error $E_T(f)$ of Trapezoidal rule is

$$\begin{aligned} E_T(f) &= I(f) - T(f) \\ &= \int_a^b f(x)dx - \int_a^b p_1(x)dx = \int_a^b R(x)dx \\ &= \int_a^b \frac{f''(\xi_x)}{2}(x-a)(x-b)dx\,. \end{aligned}$$

Using the integral mean value theroem,

$$E_T(f) = \frac{f''(c)}{2}\left[-\frac{1}{6}(b-a)^3\right] = \frac{-f''(c)}{12}(b-a)^3, \qquad c \in [a,b]\,.$$

Let $M_2 = \max_{a \le x \le b} |f'''(x)|$. Then

$$|E_T| \le \frac{M_2}{12}(b-1)^3\,. \tag{3.2}$$

Composite Trapezoidal Formula

If $(b-a)$ is not sufficiently small, we break the interval into a sum of integrals over small subintervals, it also just means that approximating $f(x)$ by a piecewise linear interpolation

$I_1(x)$ and take it as integrand.

$$\begin{aligned} I(f) &= \int_a^b f(x)dx \\ &\approx \int_a^b I_1(x)dx = \sum_{j=1}^{n} \int_{x_j-1}^{x_j} I_1(x)dx \\ &= \sum_{j=1}^{n} \frac{(x_j - x_{j-1})}{2}[f(x_{j-1} + f(x_j)] = I_{CT}(f)\,. \end{aligned}$$

For equally spaced x_j with $h = \dfrac{b-a}{n}$,

$$I_{CT}(f) = \frac{h}{2}\left[f(x_0) + f(x_n) + 2\sum_{j=1}^{n-1} f(x_j)\right]\,.$$

Error of Composite Trapezoidal Rule

Let M_2 be a bound for $|f''(x)|$ over the interval $[a, b]$. Then the error of Composite Trapezoidal Rule, E_{CT}, is

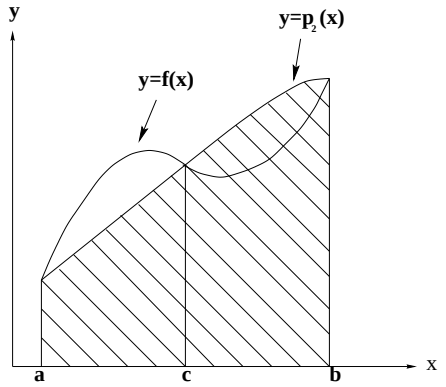
$$\begin{aligned} |E_{CT}| &= \left|\int_a^b f(x)dx - \int_a^b I_1(x)dx\right| \\ &= \left|\sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} \frac{f''(\xi_j)}{2}(x - x_{j-1})(x - x_j)dx\right| \\ &\le \sum_{j=1}^{n} \frac{M_2}{12}(x_j - x_{j-1})^3\,. \end{aligned}$$

If x_j are equally spaced, then

$$|E_{CT}| \le \frac{M_2}{12}(b-a)h^2, \qquad h = \frac{b-a}{n}\,. \tag{3.3}$$

3.1.2 Simpson's Rule

To improve upon the simple trapezoidal rule, we use a quadratic interpolation $p_2(x)$ to approximate $f(x)$ on $[a, b]$. Let $c = \dfrac{a+b}{2}$



then

$$p_2(x) = \frac{(x-c)(x-b)}{(a-c)(a-b)}f(a) + \frac{(x-a)(x-b)}{(c-a)(c-b)}f(c) + \frac{(x-a)(x-c)}{(b-a)(b-c)}f(b)$$

and

$$\begin{aligned} I(f) &= \int_a^b f(x)dx \approx \int_a^b p_2(x)dx \\ &= \frac{(b-a)}{6}[f(a) + 4f(c) + f(b)] = S(f)\,. \end{aligned} \tag{3.4}$$

We can derive the following error $E_S(f)$ of Simpson's rule,

$$E_S(f) = I(f) - S(f) = \frac{-f^{(4)}(\xi)}{2880}(b-a)^5, \qquad \xi \in [a,b]\,.$$

Let M_4 be a bound for $|f^{(4)}(x)|$ over the interval $[a,b]$. We have

$$|E_S(f)| \le \frac{M_4}{2880}(b-a)^5\,. \tag{3.5}$$

Composite Simpson's Rule

Approximating $f(x)$ by a piecewise quadratic interpolation $I_2(x)$ and taking it as integrand, we have

$$\begin{aligned} I(f) &\approx \int_a^b I_2(x)dx = \sum_{j=1}^{n}\int_{x_{j-1}}^{x_j} I_2(x)dx \\ &= \sum_{j=1}^{n}\left(\frac{x_j - x_{j-1}}{6}\right)\left[f(x_{j-1}) + 4f\left(\frac{x_{j-1}+x_j}{2}\right) + f(x_j)\right] \\ &= I_{CS}(f) \end{aligned}$$

For equally spaced x_j with $h = \dfrac{b-a}{n}$

$$I_{CS}(f) = \frac{h}{6}\left[f(x_0) + f(x_n) + 2\sum_{j=1}^{n-1} f(x_j) + 4\sum_{j=1}^{n-1} f\left(\frac{x_{j-1}+x_j}{2}\right)\right]$$

The error of Composite Simpson's Rule, E_{CS}, is

$$E_{CS} = \sum_{j=1}^{n} \frac{-f^{(4)}(\xi_j)}{2880}(x_j - x_{j-1})^5, \qquad \xi_j \in [x_{j-1}, x_j]$$

$$= \frac{-f^{(4)}(\xi)}{2880}(b-a)h^4, \qquad \xi \in [a,b]$$

For equally spaced x_j,

$$|E_{CS}| \le \frac{M_4}{2880}(b-a)h^4\,. \tag{3.6}$$

where M_4 is a bound of $|f^{(4)}(x)|$ on $[a,b]$.

3.1.3 Romberg Integration

Let $T(h)$ denote the Composite Trapezoidal Approximation of the integral $I = \int_a^b f(x)dx$ with step size h. If $f(x)$ is sufficiently smooth, say $f(x) \in C^{2m+1}[a,b]$, then $T(h)$ has an asymptotic expansion called Euler-MacLaurin formula as following,

$$T(h) = I + c_1h^2 + c_2h^4 + \cdots + c_mh^{2m} + O(h^{2m+1})$$

where c_i depend on f and interval $[a,b]$, but are independent of h. Define a new approximation

$$T_1(h) = \frac{1}{3}\left[4T(\frac{h}{2}) - T(h)\,.\right]$$

From the Euler-MacLaurin formula, we have again an expansion

$$T_1(h) = I + c_2'h^4 + c_3'h^6 + \cdots + c_m'h^{2m} + O(h^{2m+1})$$

$T_1'(h)$ is a fourth-order approximation. The process can be continued by combining $T_1(h)$ and $T_1(h/2)$ in a similar fashion to eliminate the h^4 term in the error for T_1. More generally, we can construct a triangular array from original approximations $T(h), T(\frac{h}{2}), T(\frac{h}{4})\cdots$ by the following formula

$$T_k\left(\frac{h}{2^{j-1}}\right) = \frac{4^kT_{k-1}(\frac{h}{2^j}) - T_{k-1}(\frac{h}{2^{j-1}})}{4^k-1} \qquad k = 1,\ldots,m \qquad j = 1,\ldots,m$$

where $T_0\left(\frac{h}{2^j}\right) = T\left(\frac{h}{2^j}\right)$

$$\begin{array}{llllll}
T(h) & & & & & \\
T(h/2) & T_1(h) & & & & \\
T(h/4) & T_1(h/2) & T_2(h) & & & \\
T(h/8) & T_1(h/4) & T(2(h/2) & T_3(h) & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
T(h/2^m) & T_1(h/2^{m-1}) & T_2(h/2^{m-2}) & T_3(h/2^{m-3}) & \cdots & T_m(h)
\end{array}$$

The i^{th} column of this array converges to the integral I at a convergent order $O(h^{2i})$. The method mentioned above is known as Romberg Integration.

3.1.4 Adaptive Quadrature Method

The composite formulas require the use of equally spaced nodes. But it is inapproximate when integrating a function on an interval that contains both regions with large functional variation. An efficient technique for this type of problem is one that can distinguish the amount of functional variation and adapt the step size to the varying requirements of the problem. Such methods are named Adaptive Quadrature Methods. The method we discuss is based on the Simpson's rule, but the technique can be modified to the other.
Let

$$S(a,b) = \frac{(b-a)}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

It is known that

$$\int_a^b f(x)dx = S(a,b) - \frac{(b-a)^5}{2880}f^{(4)}(\xi), \qquad \xi \in [a,b]$$

Now suppose that we wish to approximate $\int_a^b f(x)dx$ to within a specified tolerance $\epsilon > 0$. The first step is to apply Simpson's rule with step size $h = (b-a)$. The next step is to determine a way to estimate the accuracy of the approximation that does not require determining $f^{(4)}(\xi)$. To accomplish this , we first apply Composite Simpson's rule to the problem with 2 subinterval and step size $h/2$. Thus, we have

$$\begin{aligned}
\int_a^b f(x)dx &= S(a,b) - \frac{f^{(4)}(\xi)}{2880}(b-a)h^4, \qquad \xi \in [a,b] \\
\int_a^b f(x)dx &= S(a,\frac{a+b}{2}) + S(\frac{a+b}{2},b) - \frac{f^{(4)}(\xi')}{2880}(b-a)(\frac{h}{2})^4 \qquad \xi' \in [a,b]\,.
\end{aligned}$$

The error estimation is derived by assuming

$$f^{(4)}(\xi) \approx f^{(4)}(\xi')\,.$$

The success of the technique depends on the accuracy of this assumption. Equating

$$\begin{aligned}
& S(a,b) - \frac{f^{(4)}(\xi')}{2880}(b-a)h^4 \\
\approx\ & S\left(a,\frac{a+b}{2}\right) + S\left(\frac{a+b}{2},b\right) - \frac{f^{(4)}(\xi')}{2880}(b-a)\left(\frac{h}{2}\right)^4
\end{aligned}$$

we obtain

$$\begin{aligned}
& \frac{f^{(4)}(\xi')}{2880}(b-a)\left(\frac{h}{2}\right)^4 \\
\approx\ & \frac{1}{15}\left\{S(a,b) - \left[S\left(a,\frac{a+b}{2}\right) + S\left(\frac{a+b}{2},b\right)\right]\right\}\,.
\end{aligned} \tag{3.7}$$

and then

$$\left|\int_a^b f(x)dx - \left[S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right)\right]\right|$$

$$\approx \frac{1}{15}\left|S(a,b) - \left[S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right)\right]\right|$$

Therefore, if

$$\left|S(a,b) - \left[S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right)\right]\right| < 15\epsilon \tag{3.8}$$

then

$$\left|\int_a^b f(x)dx - \left[S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right)\right]\right| < \epsilon\,.$$

When the inequality (3.8) does not hold, the error estimation procedure is applied individually to the subintervals $[a,(a+b)/2]$ and $[(a+b)/2,b]$ to determine if the approximation to the integral on each subinterval is within a tolerance of $\epsilon/2$. If so, sum the approximations to produce an approximation to $\int_a^b f(x)dx$ within the tolerance ϵ. If the approximation on one of the subintervals fails to be within the tolerance $\epsilon/2$, that subinterval is itself subdivided and each of its subintervals analyszed to determine if the approximation on that subinterval is accurate to within $\epsilon/4$. This Halving Procedure is continued until each portion is within the required tolerance.

Remark: For more reliability, we set 10ϵ or ϵ rather than the 15ϵ figure shown in inequality (3.8). This smaller bound is chosen to compensate for error in the assumption $f^{(4)}(\xi') = f^{(4)}(\xi)$.

Problem 3.1 Apply the trapezoid and Simpson's rules to the function $f(x) = x^4$ on the interval $[0, 1]$. Compare the actural error in the approximations to the bounds given by (3.2) and (3.5).

Problem 3.2 Based on the bound (3.3), how small would h need to be in order to guarantee an error no larger than 10^{-6} in the composite trapezoid rule approximation for $f(x) = x^4$ on $[0, 1]$. How small for the composite Simpson's rule?

Problem 3.3 Write a computer program to carry out the composite trapezoid and Simpson's rule for an "arbitrary" function on the interval $[a, b]$ and with an arbitrary subdivision of $[a, b]$. Test your program on $f(x) = x^4$ on $[0, 1]$ and actural h needed, in the case of an equal subdivision, to achieve an error of less than 10^{-6} with the composite trapezoid rule. Do the same for $f(x) = e^{-x^2}$.

3.2 Initial Value Problems

Many problems in applied mathematics lead to Ordinary Differential Equations (ODEs). In the simplest case one seeks a differentiable function $y = y(x)$ of one real variable x, whose derivative $y'(x)$ is to satisfy an equation of the form $y'(x) = f(x, y(x))$, or more briefly,

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

one then speaks of an ordinary differential equation of first order. In general there are infinitely many different functions y which are solutions of an ODE. Through additional requirements one can single out certain solution from the set of all solutions. Thus, in an Initial-Value Problem, one seeks a solution y of an ODE which for given (x_0, y_0) satisfies an Initial Condition of the form

$$y(x_0) = y_0 \,.$$

System of ODEs

More generally, one also considers system of n ordinary differential equations

$$\begin{cases} y_1'(x) = f_1(x, y_1(x), y_2(x), \ldots, y_n(x)) \\ y_2'(x) = f_2(x, y_1(x), y_2(x), \ldots, y_n(x)) \\ \qquad\qquad \vdots \\ y_n'(x) = f_n(x, y_1(x), y_2(x), \ldots, y_n(x)) \end{cases}$$

with initial conditions

$$y_1(x_0) = y_{10}, \quad y_2(x_0) = y_{20}, \quad \ldots \quad y_n(x_0) = y_{n0} \,.$$

High order ODEs

There are ODEs of m^{th} order of the form

$$y^{(m)}(x) = f(x,\ y(x),\ y'(x), \ldots, y^{(m-1)}(x))$$

The initial conditions should be

$$y^{(i)}(x_0) = y_{i0} \qquad\qquad i = 0, 1, \ldots, m-1$$

Transformation

By introducing auxiliary functions

$$z_1(x) = y(x), \quad z_2(x) = y'(x), \ \ldots \ z_m(x) = y^{(m-1)}(x)$$

The higher order ODE can be transformed into an equivalent system of first-order differential equations.

$$\begin{cases} z_1' &= z_2 \\ z_2' &= z_3 \\ \vdots & \\ z_{m-1}' &= z_m \\ z_m' &= f(x, z_1, z_2, \ldots, z_m) \end{cases}$$

And then the initial conditions are

$$z_1(x_0) = y_{00}, \quad z_2(x_0) = y_{10}, \ \ldots \ z_m(x_0) = y_{m-1,0}$$

Example : Orbit of a Planet. Let m, M denote the mass of a planet and the sun respectively. According to the Gravitation Law, there is a gravitation

$$\vec{F} = -k\frac{Mm}{|\vec{r}|^2}\,\frac{\vec{r}}{|\vec{r}|}$$

acting on the planet, where $\vec{r}$ is the vector connecting the sun to the planet.

Put the sun on the origin of coordinates, then $\vec{r} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ represents the displacement of the planet, and satisfies the Newton Second Law of Motion:

$$\begin{aligned} m\,\frac{d^2\vec{r}}{dt^2} &= \vec{F} \\ &= -k\frac{Mm}{|\vec{r}|^3}\,\vec{r} \end{aligned}$$

Hence, we obtain an ODE system of second order

$$\begin{cases} x''(t) = -kM(x^2+y^2)^{-\frac{3}{2}}x\,, \\ y''(t) = -kM(x^2+y^2)^{-\frac{3}{2}}y\,. \end{cases} \tag{3.9}$$

The initial conditions are of the form

$$\begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases} \qquad \begin{cases} x'(t_0) = v_{x0} \\ y'(t_0) = v_{y0} \end{cases}$$

Example : Trajectory Consider the trajectory of a shell. Assume the resistance of air is proportional to the square of the velocity. Thus the trajectory of a shell is governed by the ODE

$$m\,\frac{d^2\vec{r}}{dt^2} = -mg\,\vec{e_2} - k\left|\frac{d\vec{r}}{dt}\right|\frac{d\vec{r}}{dt}\,. \tag{3.10}$$

where $\vec{r}(t)$ represents the displacement of the shell with mass m and $\vec{e_2}$ the unit vector on the y-axis (the direction perpendicular to the earth).

3.2.1 Euler's Method

For solving the initial value problem

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

the simplest method ariese by approximating $y'(x)$ by the finite difference quotient

$$y'(x) \approx \frac{1}{h}[y(x+h) - y(x)]$$

Using this approximation at the initial point, i.e. for $x = x_0$, gives

$$y(x_0 + h) \approx y(x_0) + h\, f(x_0,\, y(x_0))$$

Let $x_1 = x_0 + h$. Since $y(x_0)$ is known from initial condition $y_0 = y(x_0)$, then $y_1 = y_0 + h\, f(x_0,\, y_0)$ gives an approximation to the solution at x_1. The proess can be continued. Let $x_k = x_0 + kh,\ k = 0, 1, \ldots$ and define the approximations y_{k+1} to the solution at x_{k+1} by

$$y_{k+1} = y_k + h\, f(x_k,\, y_k), \qquad k = 0, 1, \ldots \tag{3.11}$$

The above algorithm is called Euler's method.

Discretization Error

Another derivation of Euler's method is by Taylor expansion.

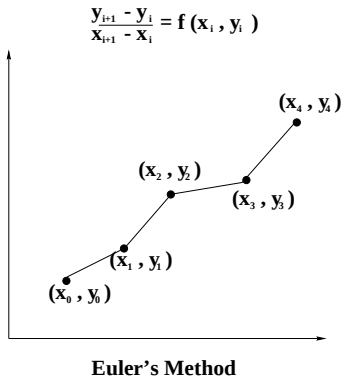
$$y(x) = y(x_k) + y'(x_k)(x - x_k) + \frac{1}{2}y''(\xi_k)(x - x_k)^2, \qquad \xi_k \text{ is between } x_k \text{ and } x\,.$$

Dropping the remainder term we have

$$y(x) \approx y(x_k) + (x - x_k) f(x_k,\, y(x_k))$$

Taking $x = x_{k+1}$ gives

$$y(x_{k+1}) \approx y(x_k) + h\, f(x_k,\, y(x_k))\,.$$



Euler's Method

It can be seen that, the remainder term in Taylor expansion is of order $O(h^2)$. It menas that if we use an accurate $y(x_k)$ to compute y_{k+1} by

$$y_{k+1} = y(x_k) + h\, f(x_k,\, y(x_k))$$

then the error of $y_{k+1} - y(x_{k+1})$ is of order $O(h^2)$. But now there is an accumulative error propagated from y_1 to y_k. Hence, the order must be lower than $O(h^2)$.

Global Discretization Error

Let $a = x_0, \quad x_N = x_0 + Nh = b$ and $x_i = a + ih, \quad i = 0, 1, \ldots$. Assume $y(x_i)$ and y_i are the exact solution and approximation obtained by Euler's method respectively. Then the Golbal Discretization Error is defined as

$$e(h) = \max_{1 \le i \le N} |y_i - y(x_i)|$$

We wish to know whether and how rapidly $e(h) \to 0$ as $h \to 0$ (Note that then $N = \dfrac{(b-a)}{h} \to \infty$).

Theorem 3.1 If the function $f(x, y)$ has a bounded partial derivative with respect to its second variable and if the solution has a bounded second derivative, then the Euler approximation converges to the exact solution as $h \to 0$ and $e(h) = O(h)$.

Proof: Let $e_k = y(x_k) - y_k, \qquad \left|\dfrac{\partial f}{\partial y}\right| \le K \quad$ and $\quad |y''| \le M_2$

$$\begin{aligned}
\text{Since} \qquad y(x_k) &= y(x_{k-1}) + h\, f(x_{k-1},\, y(x_{k-1})) + \frac{h^2}{2} y''(\xi_{k-1}), \\
y_k &= y_{k-1} + h\, f(x_{k-1},\, y_{k-1}), \\
\text{then} \qquad y(x_k) - y_k &= y(x_{k-1}) - y_{k-1} + h[f(x_{k-1},\, y(x_{k-1})) - f(x_{k-1},\, y_{k-1})] \\
&\quad + \frac{h^2}{2} y''(\xi_k).
\end{aligned}$$

Because $[f(x_{k-1},\, y(x_{k-1})) - f(x_{k-1},\, y_{k-1})] = \dfrac{\partial f}{\partial y}(x_{k-1},\, \zeta_{k-1})[y(x_{k-1}) - y_{k-1}]$,
where ζ_{k-1} is between y_{k-1} and $y(x_{k-1})$,

we have
$$\begin{aligned}
|e_k| &\le |e_{k-1}| + h\, K |e_{k-1}| + \frac{M_2}{2} h^2 \\
&= (1 + h\, K)|e_{k-1}| + \frac{M_2}{2} h^2
\end{aligned}$$

Apply the above inequality recursively,

$$\begin{aligned}
|e_k| &\le (1+h\,K)\left\{(1+h\,K)|e_{k-2}| + \frac{M_2}{2}h^2\right\} + \frac{M_2}{2}h^2 \\
&= (1+h\,K)^2|e_{k-2}| + (1+h\,K)\frac{M_2}{2}h^2 + \frac{M_2}{2}h^2 \\
&\le (1+h\,K)^3|e_{k-3}| + (1+h\,K)^2\frac{M_2}{2}h^2 + (1+h\,K)\frac{M_2}{2}h^2 + \frac{M_2}{2}h^2 \\
&\le \cdots \\
&\le (1+h\,K)^k|e_0| + \left[(1+h\,K)^{k-1} + (1+h\,K)^{k-2} + \cdots + 1\right]\frac{M_2}{2}h^2 \\
&= \frac{M_2}{2}h^2\sum_{j=0}^{k-1}(1+h\,K)^j \\
&= \frac{(1+h\,K)^k - 1}{h\,K}\,\frac{M_2}{2}h^2 \\
&\le \frac{(1+h\,K)^N - 1}{h\,K}\,\frac{M_2}{2}h^2
\end{aligned}$$

Since $(1+h\,K) \le e^{h\,K}$ then $(1+h\,K)^N \le e^{N\,h\,K} = e^{(b-a)K}$

Hence, for all $0 \le k \le N$

$$|e_k| \le C\,h, \qquad \text{where } C = \frac{(e^{(b-a)K} - 1)M_2}{2K}\,.$$

The above theorem implies that Euler's method is of first-order accuracy. If we halve the step size, h, it is expected that the error will decrease by about a factor of 2.

Example : Consider the initial problem

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

The exact solution is $y(x) = e^x$. We compute the solution at $x = 1$ by Euler's method using $h = 1,\ 1/2,\ 1/4,\ 1/8,\ 1/16$. It is seen from Table 3.1 that the error decreases by about a factor of 2 when the mesh size h decreases.

Higher-Order Methods by Taylor Expansion

The conceptually simplest approach to higher-order methods is to use more terms in the Taylor expansion. Compared with Euler's method, one more term is taken so that

$$y(x_{k+1}) \approx y(x_k) + h\,y'(x_k) + \frac{h^2}{2}y''(x_k)$$

h	Computed Value	Error	Error Ratio
1	2.000	0.718	
1/2	2.250	0.468	0.65
1/4	2.441	0.277	0.59
1/8	2.566	0.152	0.55
1/16	2.638	0.080	0.53

Table 3.1: *Error in Euler's method.*

where the remainder of $O(h^3)$ has been dropped.
As before, $y'(x_k)$ can be replaced by $f(x_k, y(x_k))$, but we now need to replace $y''(x_k)$ also. It follows from

$$y''(x) = \frac{d}{dx} f(x, y(x)) = f_x(x, y) + f_y(x, y) y'(x)\,,$$

that

$$y''(x_k) = f_x(x_k, y(x_k)) + f_y(x_k, y(x_k)) \cdot f(x_k, y(x_k))\,.$$

Using this to replace $y''(x_k)$ leads to the method

$$y_{k+1} = y_k + h\, f(x_k, y_k) + \frac{h^2}{2}[f_x(x_k, y_k) + f_y(x_k, y_k) f(x_k, y_k)]$$

which can be shown the global discretization error of $O(h^2)$.

3.2.2 Runge-Kutta Methods

Instead of computing the partial derivatives of $f(x, y)$, we could also obtain highter-order methods by making more evaluations of $f(x, y)$ at each step. This class of methods known as Runge-Kutta methods are more efficient than the above by Taylor expansion. The simplest such method is

$$y_{k+1} = y_k + \frac{h}{2}[f(x_k, y_k) + f(x_{k+1}, y_k + h\, f(x_k, y_k)]\,. \tag{3.12}$$

where we have replaced $f(x_k, y_k)$ in Euler's method by an average of $f(x, y)$ evaluated at (x_k, y_k) and $(x_{k+1}, y_k + h\, f(x_k, y_k))$, the later one is just the point obtained by Euler's method. For the method, it can be shown that $e(h) = O(h^2)$, so it is called the second-order Runge-Kutta method.

The most famous of the Runge Kutta methods is the fourth-order method given by

$$\begin{cases} y_{k+1} &= y_k + \dfrac{h}{6}(F_1 + 2F_2 + 2F_3 + F_4) \\ F_1 &= f(x_k, y_k), \qquad F_2 = f(x_k + \dfrac{h}{2},\, y_k + \dfrac{h}{2} F_1) \\ F_3 &= f(x_k + \dfrac{h}{2},\, y_k + \dfrac{h}{2} F_2), \qquad F_4 = f(x_{k+1},\, y_k + hF_3) \end{cases} \tag{3.13}$$

Here, the $f(x_k, y_k)$ in Euler's method has been replaced by a weighted average of f evaluated at four different points. Methods of order p require $p+1$ evaluations of f for $5 \leq p \leq 7$ and $p+2$ evaluations for $p \geq 8$. This is in contrast to only p evaluations if $p \leq 4$.

It is interesting to note that if we apply the above mentioned methods to the initial problem

$$\begin{cases} y' & = f(x) \\ y(a) & = 0 \end{cases}$$

then, Euler's method is the composite rectangle rule; the second-order Runge-Kutta method is the composite trapezoidal rule; and the fourth-order Runge-Kutta method is the composite Simpson's rule.

Example : Consider again the initial problem

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

We now compute the solution at $x = 1$ by fourth-order Runge-Kutta method. The exact solution $y(1) = e = 2.71828182846$.

h	Computed Value	Error	Error Ratio
1	2.70833333333	0.99485^{-2}	
1/2	2.71734619141	0.93564^{-3}	0.094
1/4	2.71820993920	0.71889^{-4}	0.077
1/8	2.71827684442	0.49840^{-5}	0.069
1/16	2.71828150034	0.32812^{-6}	0.066
1/32	2.71828180741	0.21048^{-7}	0.064
1/64	2.71828182713	0.13327^{-8}	0.063

Table 3.2: *Error in Fourth-Order Runge-Kutta Method.*

Rounding Error

As the error estimate shows, the discretization error goes to zero as h goes to zero. However, the smaller h is, the more steps is needed to go to attain a fixed point and then the larger will be the rounding error. In practice, for a fixed word length in the computer arithmetic, there will be a size of h below which the rounding error will become the dominant contribution to the overall error. This minimum h is very difficult to ascertain in advance but usually can be checked out from some numerical phenomenon, for example, the violation of monotone of the data sequence computed. The minimum h will vary from method to method and problem to problem.

Forward Euler, dy/dt=y, y(0)=1
solution at t=1 (single precision)

h	y(t)	yh	y-yh
.50000000E+00	.27182817E+01	.22500000E+01	.46828175E+00
.25000000E+00	.27182817E+01	.24414063E+01	.27687550E+00
.12500000E+00	.27182817E+01	.25657845E+01	.15249729E+00
.62500000E-01	.27182817E+01	.26379285E+01	.80353260E-01
.31250000E-01	.27182817E+01	.26769896E+01	.41292191E-01
.15625000E-01	.27182817E+01	.26973457E+01	.20936012E-01
.78125000E-02	.27182817E+01	.27077391E+01	.10542631E-01
.39062500E-02	.27182817E+01	.27129931E+01	.52886009E-02
.19531250E-02	.27182817E+01	.27156358E+01	.26459694E-02
.97656250E-03	.27182817E+01	.27169552E+01	.13265610E-02
.48828125E-03	.27182817E+01	.27176132E+01	.66852570E-03
.24414063E-03	.27182817E+01	.27179534E+01	.32830238E-03
.12207031E-03	.27182817E+01	.27181191E+01	.16260147E-03
.61035156E-04	.27182817E+01	.27181976E+01	.84161758E-04
.30517578E-04	.27182817E+01	.27182360E+01	.45776367E-04
.15258789E-04	.27182817E+01	.27182436E+01	.38146973E-04
.76293945E-05	.27182817E+01	.27182052E+01	.76532364E-04
.38146973E-05	.27182817E+01	.27178402E+01	.44155121E-03
.19073486E-05	.27182817E+01	.27170022E+01	.12795925E-02
.95367432E-06	.27182817E+01	.27137330E+01	.45487881E-02
.47683716E-06	.27182817E+01	.26949396E+01	.23342133E-01

4th order Runge-Kutta, dy/dt=y, y(0)=1
solution at t=1 (single precision)

h	y(t)	yh	y-yh
.10000000E+01	.27182817E+01	.27083335E+01	.99482536E-02
.50000000E+00	.27182817E+01	.27173462E+01	.93555450E-03
.25000000E+00	.27182817E+01	.27182102E+01	.71525574E-04
.12500000E+00	.27182817E+01	.27182767E+01	.50067902E-05
.62500000E-01	.27182817E+01	.27182813E+01	.47683716E-06
.31250000E-01	.27182817E+01	.27182822E+01	-.47683716E-06
.15625000E-01	.27182817E+01	.27182820E+01	-.23841858E-06
.78125000E-02	.27182817E+01	.27182834E+01	-.16689301E-05
.39062500E-02	.27182817E+01	.27182829E+01	-.11920929E-05
.19531250E-02	.27182817E+01	.27182825E+01	-.71525574E-06

```
.97656250E-03   .27182817E+01   .27182789E+01   .28610229E-05
.48828125E-03   .27182817E+01   .27182808E+01   .95367432E-06
.24414063E-03   .27182817E+01   .27182782E+01   .35762787E-05
.12207031E-03   .27182817E+01   .27182817E+01   .00000000E+00
.61035156E-04   .27182817E+01   .27182791E+01   .26226044E-05
.30517578E-04   .27182817E+01   .27182746E+01   .71525574E-05
.15258789E-04   .27182817E+01   .27182662E+01   .15497208E-04
.76293945E-05   .27182817E+01   .27182157E+01   .66041946E-04
.38146973E-05   .27182817E+01   .27178450E+01   .43678284E-03
.19073486E-05   .27182817E+01   .27170050E+01   .12767315E-02
```

3.2.3 Multistep Methods

Another approach to higher-order methods utilizies information already computed and does not require additional evalutaions of $f(x, y)$. One of the simplest such methods is

$$y_{k+1} = y_k + \frac{h}{2}[3f(x_k, y_k) - f(x_{k-1}, y_{k-1})] \tag{3.14}$$

for which $e(h) = O(h^2)$, and is known as the second-order Adams-Bashforth method. Note that the method requires only the evaluation of $f(x_k, y_k)$ at each step, the value $f(x_{k-1}, y_{k-1})$ being known from the previous step.

We now consider the general construction of Adams-Bashforth methods. Let y_k, $y_{k-1}, \ldots, y_{k-N}$ be the computed approximations to the solution at x_k, $x_{k-1}, \ldots, x_{k-N}$. Let $f_i = f(x_i, y_i)$ and let $p(x)$ be the interpolating polynomial of degree N that satisfies

$$p(x_i) = f_i, \qquad\qquad i = k, k-1, \ldots, k-N\,.$$

We may then consider $p(x)$ to be an approximation to $f(x, y(x))$. Since the solution $y(x)$ satisfies

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} y'(x)dx = \int_{x_k}^{x_{k+1}} f(x, y(x))dx \approx \int_{x_k}^{x_{k+1}} p(x)dx$$

then we obtain the so-called Adams-Bashforth (AB) methods as following

$$y_{k+1} = y_k + \int_{x_k}^{x_{k+1}} p(x)dx\,.$$

The following lists a few special cases of the Adams-Bashforth methods:

- $N = 0$: $p(x) = f_k$ gives Euler's method.
- $N = 1$: $p(x) = p_1(x) = f_k - \dfrac{(x - x_k)}{h}(f_k - f_{k-1})$, which leads to

  $$y_{k+1} = y_k + \frac{h}{2}(2f_k - f_{k-1}),$$

  is second-order Adams-Bashforth method mentioned above.

- $N = 2$: $p_2(x) = p_1(x) + \dfrac{(x - x_k)(x - x_{k-1})}{2h^2}(f_k - 2f_{k-1} + f_{k-2})$, which leads to

$$y_{k+1} = y_k + \frac{h}{12}(23f_k - 16f_{k-1} + 5f_{k-2})$$

  is the third-order Adams-Bashforth method.

- $N = 3$: $p_3(x) = p_2(x) - \dfrac{(x - x_k)(x - x_{k-1})(x - x_{k-2})}{3!h^3}(f_k - 3f_{k-1} + 3f_{k-2} - f_{k-3})$, which leads to

$$y_{k+1} = y_k + \frac{h}{24}(55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3})$$

  is a fourth-order AB method.

In principle, we can continue the preceding process to obtain Adams-Bashforth methods of arbitrarily high-order, but the formulas become increasingly complex as N increases. The Adams-Bashforth methods are called multistep methods since prior values of the data are used. This is in contrast to the Runge-Kutta methods which use no prior data and are called one-step methods.

Starting values

Multistep methods can not start by theirselves. For example, consider the fourth-order Adams-Bashforth method. The initial value y_0 is given, but for $k = 0$, the information is needed at x_{-1}, x_{-2}, x_{-3} which doesn't exist. The method needs "help" getting started. We can not use the fourth-order multistep method until $k \geq 3$.

A common policy is to use a one-step method, such as a Runge-Kutta method of the same order of accuracy at some starting steps.

Comparison

Adams-Bashforth methods of arbitrary order require only one evaluation of $f(x, y)$ at each step, the "cost" is lower than that of Runge-Kutta methods. But they can not self-start and need the help of Runge-Kutta methods. Runge-Kutta methods are much easier to change step-size, hence they are more suitable for use in an adaptive algorithm.

Implicit and Predictor-Corrector Methods

The Adams-Bashforth methods can be unstable due to the fact they are obtained by integrating the interpolating polynomial outside the interval of the data that defines the polynomial. This can be remedied by using data at x_{k+1} to determine the interpolating polynomial.

Consider $f(x, y(x))$ approximated by $p(x)$ which is the linear function on $[x_k, x_{k+1}]$ that interpolates (x_k, f_k) and (x_{k+1}, f_{k+1}), then from

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} y'(x)dx = \int_{x_k}^{x_{k+1}} f(x, y(x))dx \approx \int_{x_k}^{x_{k+1}} p(x)dx$$

we obtain a formula

$$y_{k+1} = y_k + \frac{h}{2}(f_{k+1} + f_k)$$

Similarly, if $p(x)$ is the cubic interpolating polynomial for the points (x_{k+1}, f_{k+1}), (x_k, f_k), (x_{k-1}, f_{k-1}) and (x_{k-2}, f_{k-2}), then the formula is

$$y_{k+1} = y_k + \frac{h}{24}[9f_{k+1} + 19f_k - 5f_{k-1} + f_{k-2}]$$

This kind of formula is called the Adams-Moulton method. The former one is of second order and the latter one is of fourth order.

Now, note that in the above formulas, f_{k+1} is not known since we need y_{k+1} to evaluate $f_{k+1} = f(x_{k+1}, y_{k+1})$, but y_{k+1} is not get known. Hence the Adams-Moulton methods define y_{k+1} only implicitly. Thus, the Adams-Moulton methods are called Implicit whereas the Adams-Bashforth methods are called Explicit since no equation needs to be solved to obtain y_{k+1} at each step.

In principle, we could attempt to solve the nonlinear algebraic equations for y_{k+1} by Newton's method or other methods, discussed in next chapter. However, another use of implicit methods is to combine an explicit formula with an implicit one to form a Predictor-Corrector method. For example, the second order Adams-Moulton methods define y_{k+1} as the solution of the following nonlinear equation

$$y_{k+1} = y_k + \frac{h}{2}[f(x_{k+1}, y_{k+1}) + f(x_k, y_k)]$$

Instead of interative method, we get a "predicted" value $y_{k+1}^{(p)}$ by the second order explicit Adams-Bashforth formula

$$y_{k+1}^{(p)} = y_k + \frac{h}{2}(3f_k - f_{k-1})$$

and then compute $f_{k+1}^{(p)} = f(x_{k+1}, y_{k+1}^{(p)})$ and substitute

$$y_{k+1} = y_k + \frac{h}{2}[f_{k+1}^{(p)} + f_k]$$

The method is entirely explicit.

A commonly used predictor-corrector method is the combination of the fourth-order explicit and implicit Adams methods:

$$\begin{cases} y_{k+1}^{(p)} &= y_k + \dfrac{h}{24}(55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}) \\ f_{k+1}^{(p)} &= f(x_{k+1}, y_{k+1}^{(p)}) \\ y_{k+1} &= y_k + \dfrac{h}{24}(9f_{k+1}^{(p)} + 19f_k - 5f_{k-1} + f_{k-2}) \end{cases}$$

The Adams-Bashforth formula is used as a "predictor" and the Adams-Moulton formula is used as a "corrector".

We could also take additional corrector steps. It just equal an iterative method to solve the nonlinear equation.

3.2.4 Numerical Solutions of Systems of ODEs

We now indicate how the methods that have been discussed can be used for systems of equations. Write the system in the vector form

$$Y'(x) = F(x, Y(x))$$

where $Y(x)$ and $F(x, y)$ represent the vectors

$$Y(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_n(x) \end{bmatrix} \qquad F(x, y) = \begin{bmatrix} f_1(x, y_1, \ldots, y_n) \\ f_2(x, y_1, \ldots, y_n) \\ \vdots \\ f_n(x, y_1, \ldots, y_n) \end{bmatrix}$$

Then, Euler's method can be written as

$$Y^{(k+1)} = Y^{(k)} + h\, F(x_k, Y^{(k)})$$

In component form, for example $n = 2$, this would be

$$\begin{aligned} y_1^{(k+1)} &= y_1^{(k)} + h\, f_1(x_k, y_1^{(k)}, y_2^{(k)}) \\ y_2^{(k+1)} &= y_2^{(k)} + h\, f_2(x_k, y_1^{(k)}, y_2^{(k)}) \end{aligned}$$

Clearly, the succinct vector notation is advantageous, especially for a large number of equations. Similiarly, the second-order Runge-Kutta method can be written in vector form as

$$Y^{(k+1)} = Y^{(k)} + \frac{h}{2}[F(x_k, Y^{(k)} + F(x_{k+1}, Y^{(k)} + h\, F(x_k, Y^{(k)})]$$

and the second-order Adams-Bash-forth method can be written as

$$Y^{(k+1)} = Y^{(k)} + \frac{h}{2}[3F(x_k, Y^{(k)}) - F(x_{k-1}, Y^{(k-1)})]$$

3.2.5 Stability

Unstable Solutions

One potential difficulty in solving initial value problems is that the solution may be unstable. Consider the second-order initial problem

$$\begin{cases} y'' = 10y' - 11y \\ y(0) = 1, \quad y'(0) = -1 \end{cases} \tag{3.15}$$

The solution is $y(x) = e^{-x}$. Now suppose the initial conditions are changes to be

$$y(0) = 1 + \epsilon, \qquad y'(0) = -1$$

Then the solution is changed correspondingly to be

$$y(x) = (1 + \frac{11}{12}\epsilon)e^{-x} + \frac{\epsilon}{12}e^{11x}$$

Therefore, for any $\epsilon > 0$, no matter how small, the second term cause the solution to tend to infinity as $x \to \infty$. In this case, we say that the solution is unstable, since arbitrarily small changes in the initial condition can produce arbitrarily large changes in the solution as $x \to \infty$.

One would also say that this problem is Ill-Conditioned; it is extremely difficult to solve numerically because rounding and discretization error will cause the same effect as changing the initial conditions and the approximate solution will tend to diverge.

Unstable Methods

We now turn to possible unstabilities in the numerical method. It is second-order accurate using central divided difference to approximate the derivative, i.e.,

$$\frac{y(x+h) - y(x-h)}{2h} = y'(x) + O(h^2)$$

This gives rise to the following numerical method

$$y_{n+1} = y_{n-1} + 2h\, f(x_n, y_n)$$

We now apply the method to the problem

$$\begin{cases} y' & = -2y + 1 \\ y(0) & = 1 \end{cases}$$

whose exact solution is

$$y(x) = \frac{1}{2}e^{-2x} + \frac{1}{2}.$$

The above solution is stable: since if the initial condition is changed to $y(0) = 1 + \epsilon$, the solution becomes

$$y(x) = (\frac{1}{2} + \epsilon)e^{-2x} + \frac{1}{2}$$

and the change is $\epsilon\, e^{-2x}$.

The mentioned method applied to the problem is

$$\begin{aligned} y_{n+1} &= y_{n-1} + 2h(-2y_n + 1) \\ &= -4h\, y_n + y_{n-1} + 2h, \end{aligned}$$

Since it is a two-step method, except $y_0 = 1$, we need to supply y_1 to start the process. Assume that we take y_1 to be the exact solution at $x = h$:

$$x = h: \quad y_1 = \frac{1}{2}e^{-2h} + \frac{1}{2}\,.$$

For any fixed $h > 0$, we will see that

$$\lim_{n\to\infty} |y_n| = \infty\,,$$

where sequence y_n is generated by the method. In contrast, for the exact solution $y(x)$, $\lim_{x\to\infty} y(x) = \frac{1}{2}$ Thus, the method although second-order accurate, exhibits unstable behaviour.

A representation of approximate y_n

$$y_n = c_1\lambda_1^n + c_2\lambda_2^n + \frac{1}{2}$$

where λ_1 and λ_2 are roots of equation

$$\lambda^2 + 4h\lambda - 1 = 0$$

$$\Rightarrow \begin{cases} \lambda_1 = \sqrt{1+4h} - 2h \\ \\ \lambda_2 = -\sqrt{1+4h} - 2h \end{cases}$$

constants c_1 and c_2 are solutions of linear equations

$$\begin{cases} c_1 + c_2 + \frac{1}{2} = y_0 = 1 \\ \\ c_1\lambda_1 + c_2\lambda_2 + \frac{1}{2} = y_1 = \frac{1}{2}e^{-2h} + \frac{1}{2} \end{cases}$$

Because $\quad |\lambda_2| > 1, \quad \lim_{n\to\infty} |y_n| = \infty\,.$

Stability of Methods

All of the multistep methods described so far are special cases of what are called linear multistep methods of the form

$$y_{k+1} = \sum_{i=1}^{m} \alpha_i y_{k+1-i} + h\sum_{i=0}^{m} \beta_i f_{k+1-i}$$

where $f_j = f(x_j, y_j)$ and m is some fixed integer. If $\beta_0 = 0$, the method is explicit, and if $\beta_0 \neq 0$, then the method is implicit.

In all of the Adams methods, $\alpha_1 = 1$ and $\alpha_i = 0,\ i > 1$; in the Adams-Bashforth methods, $\beta_0 = 0$; in the Adams-Moulton methods, $\beta_0 \neq 0$. Associated with a multistep method is the polynomial

$$p(\lambda) = \lambda^m - \alpha_1\lambda^{m-1} \cdots \alpha_{m-1}\lambda - \alpha_m$$

The method is stable provided that all roots λ_i of $p(\lambda)$ satisfy $|\lambda_i| \leq 1$ and only one $|\lambda_i| = 1$.

For a m-step Adams methods, $p(\lambda) = \lambda^m - \lambda^{m-1}$, so one root is 1 and other $(m-1)$ roots are zero and hence they are stable. It is possible to extend the above framework so as to include Runge-Kutta methods, and these methods are also stable.

For the method

$$y_{n+1} = y_{n-1} + 2h\, f(x_n, y_n)$$

the polynomial is

$$p(\lambda) = \lambda^2 - 1$$

with roots ± 1; hence the method is not stable.

It should be noted that from the stability theory, a stable method will give accurate results for sufficiently small h. But, on the other hand a stable method can exhibit unstable behaviour if h is too large. Although, in principle, h can be taken sufficiently small to overcome this difficulty, the computing time may become prohibitive. This is the situation with ODEs that are known as Stiff.

Stiff Equations

An ODE is called stiff if its solution consists of a steady-state part that does not grow significantly, together with a transient part that decays rapidly to zero. Stiff problems occur in a wide variety of applications, including the study of spring and damping systems, the analysis of control systems, and problems in chemical kinetic.

As a simple example, consider the problem

$$y' = -100y + 100 \qquad y(0) = 2. \tag{3.16}$$

The exact solution of the problem is

$$y(x) = e^{-100x} + 1$$

It is clear that the solution is stable since if we change the initial condition to $y(0) = 2 + \epsilon$, then the solution changes by $\epsilon\, e^{-100x}$. Euler's method applied to this problem is

$$\begin{aligned} y_n &= y_{n-1} + h(-100y_{n-1} + 100) \\ &= (1 - 100h)y_{n-1} + 100h \end{aligned}$$

Applying the corresponding formulas for $y_{n-1}, y_{n-2}, \ldots y_1$, it is possible to obtain

$$y_n = (1 - 100h)^n + 1 \tag{3.17}$$

Note that $y(x)$ decrease very rapidly from $y(0) = 2$ to its limiting value of 1; for example, $y(0.1) \approx 1 + 5 \times 10^{-5}$. Initially, therefore, we expect to require a small step size h to compute the solution accurately. Beyond, say, $x = 0.1$, the solution varies slowly, so intuitively, we would expect to obtain sufficient accuracy using a relatively large h. But this is not correct for the Euler's method. We see from the formula (3.17) that $|1 - 100h| > 1$ when $h > 0.02$ and the approximation y_n grows at each step and shows an unstable behaviour. The quantity $(1 - 100h)^n$ is a good approximation for very small h, but becomes a poor approximation as h becomes as large as 0.02. Even though the exponential term e^{-100x} contributes virtually nothing to the solution after $x = 0.1$, Euler's method still requires that we approximate it sufficiently accurately to maintain stability. This is the typical problem with stiff equations: the solution contains a component (the transient part that decays rapidly to zero) that contributes very little to the solution, but the used methods require that it be approximated accurately.

The general approach to stiff problems is to use implicit methods. The simplest implicit method is

$$y_{n+1} = y_n + h\, f(x_{n+1}, y_{n+1})$$

which is known as the Backward Euler Method. It is of the same form as Euler's method except that f is evaluated at (x_{n+1}, y_{n+1}) rather than at (x_n, y_n), hence, the method is implicit. Using this method to the discussed problem, we obatin

$$y_{n+1} = y_n + h(-100y_{n+1} + 100)$$

which can be put in the form

$$y_{n+1} = \frac{(y_n + 100h)}{(1 + 100h)}$$

From the formula and initial condition, we obtain

$$y_n = \frac{1}{(1 + 100h)^n} + 1$$

For this approximation, we can see that there is no unstable behaviour regardless of the size of h.

The above is a simple example for demonstration. In practice, stiff problems are usually nonlinear equations or systems of nonlinear equations. Hence when using implicit methods, the solution of a system of nonlinear equations would be needed at each step. This is costly in computer time, but the effective handling of stiff equations requires that some kind of implicitness be brought into the numerical method.

Problem 3.4 Transform the ODEs in (3.9) and (3.10) into first order ODE systems.

Problem 3.5 Apply Euler's method (3.11) to the initial value problem $y' = -y, 0 \leq x \leq 1, y(0) = 1$, with $h = 0.25$. Compare your answers to the exact solution $y(x) = e^{-x}$. Repeat for $h/2$ and $h/4$. Next, apply second-order Runge-Kutta method (3.12) to this problem and compare with Euler's method.

Problem 3.6 Write a computer program to carry out the second-order Adams-Bashforth method (3.14). Use the second-order Runge-Kutta method to supply the missing starting value y_1. Apply your program to the problems of Excercise 3.5 and compare your results with the Euler and second-order Runge-Kutta methods.

Problem 3.7 By letting $z = y'$, show that the problem (3.15) is equivalent to the first-order system $y' = z, z' = 10z + 11y$, with initial condition $y(0) = 1$ and $z(0) = -1$. Attempt to solve this system numerically by any of the methods of this chapter and discuss your results.

Problem 3.8 Consider the method

$$y_{n+1} = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}),$$

which is known as *Milne's method*. Discuss whether this method is stable and strongly stable.

Problem 3.9 Write a program to carry out Euler's method on (3.16) for different values of h both less than and greater than 0.02. Discuss your results.

Problem 3.10 The function $y(x) = e^{-x}$ is the solution of the problem

$$y'' = y, \qquad y(0) = 1, \quad y'(0) = -1.$$

Is this solution stable?

3.3 Boundary Value Problems

Up to this point, we have considered numerical methods for solving only initial value problems for ODEs. For such problems, conditions on the solution of the differential equation are specified at a single point, called the initial point.

We now consider problems in which the unknown solution has conditions imposed on it at more than one point. Such problems for ODEs are called boundary value problems, or BVPs for short. Consider the second-order equation

$$v''(x) = f(x, v(x), v'(x)), \qquad a \le x \le b$$

Because it is second-order, two additional conditions are required. The simplest possibility is to prescribe the solution vaules at the endpoints:

$$v(a) = \alpha, \qquad v(b) = \beta$$

3.3.1 Finite Difference Approximation

We first consider the special case of linear problem

$$\begin{cases} v''(x) = c(x)v(x) + d(x), & a \le x \le b \\ \\ v(a) = \alpha, & v(b) = \beta. \end{cases}$$

We will assume that $c(x) \ge 0$ for $a \le x \le b$; this is a sufficient condition for the problem to have a unique solution. Let

$$h = \frac{b-a}{n+1}, \qquad x_i = a + ih, \qquad i = 0, 1, \ldots n, n+1.$$

Then $x_0 = a$ and $x_{n+1} = b$ are boundary points, and $x_1, x_2, \ldots, x_n$ are interior grid points. Using central differnece to approximate $v''(x)$, we obtain

$$\begin{aligned} v''(x_i) &\approx \frac{1}{h}\left[v'(x_i + \frac{h}{2}) - v_i'(x - \frac{h}{2})\right] \\ v'(x_i + \frac{h}{2}) &\approx \frac{1}{h}[v(x_i + h) - v(x_i)] \\ v'(x_i - \frac{h}{2}) &\approx \frac{1}{h}[v(x_i) - v(x_i - h)] \end{aligned}$$

then

$$v''(x_i) \approx \frac{1}{h^2}[v(x_i + h) - 2v(x_i) + v(x_i - h)].$$

By the equally spaced grid points, the finite differnece approximations are then

$$v''(x_i) = \frac{1}{h^2}[v(x_{i+1}) - 2v(x_i) + v(x_{i-1})]$$

Let v_i $(i = 1, \ldots, n)$ be approximates of $v(x_i)$, we obtain the following finite difference equations

$$\frac{1}{h^2}[v_{i+1} - 2v_i + v_{i-1}] - c_i v_i = d_i, \qquad i = 1, 2, \ldots, n$$

or

$$-v_{i+1} + (2 + c_i h^2)v_i - v_{i-1} = -h^2 d_i, \qquad i = 1, \ldots, n$$

where $c_i = c(x_i)$, $d_i = d(x_i)$, $v_0 = \alpha$ and $v_{n+1} = \beta$. The system of equations in the n unknowns $v_1, v_2, \ldots, v_n$ written in matrix-vector form is

$$\begin{bmatrix} 2+c_1h^2 & -1 & & & \\ -1 & 2+c_2h^2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2+c_nh^2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} -h^2d_1 + \alpha \\ -h^2d_2 \\ \vdots \\ \vdots \\ -h^2d_n + \beta \end{bmatrix}$$

The coefficient matrix has many desirable properties. It is symmetric and also positive definite (hence nonsingular and has a unique solution). It is tridiagonal and diagonally dominant, hence the solution by LU decomposition is stable.

Equations with First Derivatives

We next consider a more general two-point boundary value problem:

$$v''(x) = \gamma(x)v'(x) + c(x)v(x) + d(x), \qquad a \le x \le b, \tag{3.18}$$

where γ, c, and d are given functions of x. The standard central difference approximation to v' is

$$v'(x) \approx \frac{1}{2h}[v(x+h) - v(x-h)], \tag{3.19}$$

and it is easy to show that the error in this approximation is $O(h^2)$. If we replace v' in (3.18) at the grid point by (3.19) and proceed as before, the numerical approximation will be:

$$\left(-1 - \frac{\gamma_i h}{2}\right) v_{i-1} + (2 + c_i h^2)v_i + \left(-1 + \frac{\gamma_i h}{2}\right) v_{i+1} = -h^2 d_i \tag{3.20}$$

or

$$r_i v_{i-1} + p_i v_i + q_i v_{i+1} = -h^2 d_i, \qquad i = 1, 2, \cdots, n\,,$$

where we have set $\gamma_i = \gamma(x_i), c_i = c(x_i)$, $d_i = d(x_i)$, and

$$p_i = 2 + c_i h^2, \quad q_i = -1 + \gamma_i h/2, \quad r_i = -1 - \gamma_i h/2\,.$$

Then we can write these equations in matrix-vector form as

$$\begin{bmatrix} p1 & q_1 & & & \\ r2 & p2 & q_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & q_{n-1} \\ & & & r_n & p_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} -h^2d_1 - r_1\alpha \\ -h^2d_2 \\ \vdots \\ -h^2d_{n-1} \\ -h^2d_n - q_n\beta \end{bmatrix}. \tag{3.21}$$

In the above, α and β are given boundary values: $v(a) = \alpha, v(b) = \beta$.

One desirable property of the coefficient matrix in (3.21) is diagonal dominance.

Definition 3.1 *A general $n \times n$ matrix $A = (a_{ij})$ is row diagonally dominant if*

$$|a_{ii} \geq \sum_{j \neq i} |a_{ij}|, \qquad i = 1, \cdots, n; \tag{3.22}$$

and column diagonally dominant if

$$|a_{ii} \geq \sum_{i \neq j} |a_{ji}|, \qquad i = 1, \cdots, n. \tag{3.23}$$

Diagonal dominance is important for a number of reasons. For example, a strictly diagonally dominant matrix is nonsingular. We will see in next chapter that diagonal dominance is a valuable property in the numerical solution of a linear system.

Theorem 3.2 *The matrix in (3.21) is diagonally dominant if*

$$c_i \geq 0, \quad |\gamma_i h| \leq 2, \qquad i = 1, \cdots, n.$$

Discretization Error

Let $v(x_i)$ be the exact solution of the BVP at x_i and v_i be the approximate computed from the finite difference system with no rounding error. Then, analogous to the definition for IVP, the global discretization error $e(h)$ is defined as follows

$$e(h) = \max \left\{ |v_i - v(x_i)|, \quad i = 1, \ldots, n, \quad h = \frac{b-a}{n+1} \right\}$$

It can be shown that for $c(x) \geq 0$,

$$e(h) = O(h^2)$$

Other Boundary Conditions

In many problems, boundary conditions on the derivative rather than the function itself may be given. For example, we may have the boundary conditions

$$v'(a) = \alpha, \qquad\qquad v'(b) = \beta$$

In this case, the values of v_0 and v_{n+1} are no longer known from the boundary conditions. Hence, there will be two additional unknowns v_0 and v_{n+1} and two more equations are needed. We approximate $v''(x)$ at $x = a$ and $x = b$ by

$$v''(a) \approx \frac{1}{h^2}[v_{-1} - 2v_0 + v_1]$$

$$v''(b) \approx \frac{1}{h^2}[v_n - 2v_{n+1} + v_{n+2}]$$

using grid points $x_{-1} = a - h$ and $x_{n+2} = a + (n+2)h$ outside the interval.

Then, by the boundary conditions

$$\begin{aligned} \alpha &= v'(a) \approx \frac{1}{2h}[v_1 - v_{-1}] \\ \beta &= v'(b) \approx \frac{1}{2h}[v_{n+2} - v_n] \end{aligned}$$

we have

$$\begin{aligned} v_{-1} &= v_1 - 2h\,\alpha \\ v_{n+2} &= v_n + 2h\,\beta \end{aligned}$$

and we can use them to eliminate v_{-1} and v_{n+2},

$$\begin{aligned} v''(a) &\approx \frac{1}{h^2}[2v_1 - 2v_0 - 2\alpha h] \\ v''(b) &\approx \frac{1}{h^2}[2v_n - 2v_{n+1} + 2\beta h] \end{aligned}$$

Hence, for the ODE

$$v''(x) = c(x)v(x) + d(x)$$

we get two equations for $x = a$ and $x = b$,

$$\begin{aligned} (2 + c_0h^2)v_0 - 2v_1 &= -2\alpha h - h^2 d_0 \\ (2 + c_{n+1}h^2)v_{n+1} - 2v_n &= 2\beta h - h^2 d_{n+1} \end{aligned}$$

In summary, for the boundary value problems

$$\begin{cases} v''(x) = c(x)v(x) + d(x), & a \le x \le b \\ v'(a) = \alpha, & v'(b) = \beta. \end{cases}$$

the finite difference equations include $(n+2)$ unknowns $v_0, v_1, \ldots, v_{n+1}$ and $(n+2)$ equations which have the following matrix-vector form

$$\begin{bmatrix} 2 + c_0h^2 & -2 & & & \\ -1 & 2 + c_2h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -2 & 2 + c_{n+1}h^2 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_n \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} -h^2 d_0 - 2\alpha h \\ -h^2 d_1 \\ \vdots \\ -h^2 d_n \\ -h^2 d_{n+1} + 2\beta h \end{bmatrix}$$

The coefficient matrix is no longer symmetric. If $c_i > 0,\ i = 0, \ldots, n+1$, the matrix is nonsingular. But if all c_i are zero, it is singular, since $e = [1, 1, \ldots, 1]^T$ satisfies $Ae = 0$, where A is the coefficient matrix. This singularity of A mirrors the non-uniqueness of solutions of the ODE itself: if $c(x) \equiv 0$ and $v(x)$ is a solution satisfying the boundary conditions: $v'(a) = \alpha\ v'(b) = \beta$, then $v(x) + \gamma$ is also a solution for any constant γ.

The boundary conditions are called Dirichlet, if only the function value is specified. The boundary conditions are called Neumann, if only the derivatives are specified. More generally, mixed boundary conditions may be given as linear combinations of both function and derivative values at both endpoints:

$$\eta_1 v(a) + \eta_2 v'(a) = \alpha, \quad \gamma_1 v(b) + \gamma_2 v'(b) = \beta$$

In this case, approximations analogous to those discussed previously would be used. Another type of boundary condition is called Periodic:

$$v(a) = v(b)\,.$$

In this case, v_0 and v_{n+1} are unknowns but since they are equal only one more equation is needed to be added.

In the approximation at x_n, we can take $v_{n+1} = v_0$:

$$v''(x_n) \approx \frac{1}{h^2}[v_{n-1} - 2v_n + v_{n+1}] = \frac{1}{h^2}[v_{n-1} - 2v_n + v_0]$$

Similarly, at x_0, we can take $v_{-1} = v_n$

$$v''(x_0) \approx \frac{1}{h^2}[v_{-1} - 2v_0 + v_1] = \frac{1}{h^2}[v_n - 2v_0 + v_1]$$

Hence, the system of finite difference equation is

$$\begin{bmatrix} 2+c_0h^2 & -1 & & & -1 \\ -1 & 2+c_1h^2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ -1 & & & -1 & 2+c_nh^2 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} -h^2d_0 \\ -h^2d_1 \\ \vdots \\ -h^2d_{n-1} \\ -h^2d_n \end{bmatrix}$$

The coefficient matrix is now symmetric but tridiagonal structures has been lost. The matrix is nonsingular if all c_i are positive, but it is singular if all c_i are zero. The singularity again reflects the ODE since if $C(x) \equiv 0$ and $v(x)$ is a solution, then $v(x) + \gamma$ is also a solution for any constant γ.

Nonlinear Problems

For simplicity, we consider the function f that contains only v and not v' and the boundary conditions that are the Dirichlet type. That means the boundary value problem is of the form

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

where f is nonlinear with respect to v. As with the linear problem, we approximate $v''(x)$ by central finite difference. This leads to the system of nonlinear equations:

$$\begin{cases} -v_{i+1} + 2v_i - v_{i-1} + h^2 f(x_i, v_i) = 0, & i = 1, \ldots, n \\ \\ v_0 = \alpha, \qquad\qquad v_{n+1} = \beta \end{cases}$$

For this kind of nonlinear system, we will solve by Newton's method to be discussed in next chapter.

3.3.2 Shooting Method

Suppose that we are now going to solve the boundary value problem

$$\begin{cases} v''(x) = f(x, v(x), v'(x)) \\ \\ v(a) = \alpha, \qquad\qquad v(b) = \beta \end{cases}$$

We consider the corresponding initial value problem

$$\begin{cases} v''(x) = f(x, v(x), v'(x)) \\ \\ v(a) = \alpha, \qquad\qquad v'(a) = s \end{cases}$$

and denote the solution $v(x; s)$
If we can find a value of s, say s^*, such that

$$v(b; s^*) = \beta$$

then it just means the solution of the IVP

$$\begin{cases} v''(x) = f(x, v(x), v'(x)) \\ \\ v(a) = \alpha, \qquad\qquad v'(a) = s^* \end{cases}$$

is the solution of our BVP. Hence, we can solve the BVP through solving a series of IVPs. Choose a value s_0 and solve numerically the initial value problem with
$v'(a) = s_0$. We denote the numerical solution at $x = b$ also by $v(b; s_0)$. We then adjust the parameter s and try again with $v'(a) = s_1$ and denote the numerical solution at $x = b$ by $v(b; s_1)$. Assume that

$$v(b; s_0) - \beta > 0$$

and

$$v(b; s_1) - \beta < 0$$

then, we try again with

$$v'(a) = s_2 = \frac{1}{2}(s_0 + s_1)$$

If

$$v(b; s_2) - \beta > 0$$

then take

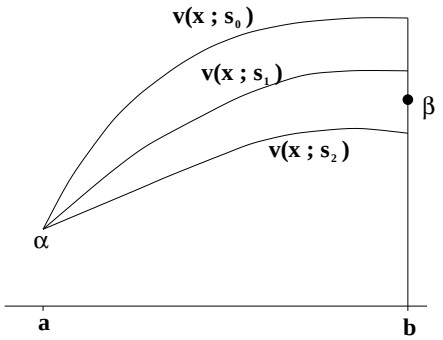
$$v'(a) = s_3 = \frac{1}{2}(s_1 + s_2)$$

otherwise take

$$v'(a) = s_3 = \frac{1}{2}(s_0 + s_2)$$

and so on.

The process is illustrated in Figure and is called the Shooting Method in analogy with an artillery gunner adjusting the angle of a gun to hit a desired target.



On the other hand, the adjustment of the parameter s can be recognized that we are attempting to solve the nonlinear equation

$$g(s) = v(b; s) - \beta = 0$$

where $g(s)$ has no explicit expression form. The procedure mentioned above is the Bisection method for finding a root of nonlinear equation. In next chapter, more efficient methods will be recommended.

Problem 3.11 Assume that the function $v(x)$ is suitably differentiable. Show that the error in the centered difference approximation (3.19) is $O(h^2)$.

Problem 3.12 Consider the two-point boundary value problem:

$$v'' = v(x) + 1, \qquad v(0) = 0, \quad v(2) = -1 + e^2.$$

Using the central difference approximation with $n = 20$ and 40 to solve it. Compare your numerical results with the exact solution $v(x) = -1 + e^x$.

Problem 3.13 Consider the two-point boundary value problem:

$$v'' + 2xv' - x^2 v = x^2, \quad v(0) = 1,\ v(1) = 0.$$

(a): Let $h = 1/4$ and explicitly write out the difference equations (3.21). **(b)**: Repeat the part (a) for the boundary conditions $v'(0) = 1, v(1) = 0$, and then $v'(0) + v(0) = 1, v'(1) + \frac{1}{2}v(1) = 0$.

Problem 3.14 Prove Theorem 3.2.

3.4 Partial Differential Equation

3.4.1 Space and Time

We now apply the techniques of the previous sections to Partial Differential Equations (PDEs) that have initial and boundary conditions.

The simplest such equation is the Heat (or Diffusion) equation:

$$\frac{\partial}{\partial t}u(t,x) = \frac{\partial^2}{\partial x^2}u(t,x), \qquad a \le x \le b, \qquad 0 \le t$$

Initial and boundary conditions are specified by

$$\begin{cases} u(0,x) = g(x), & a \le x \le b \\ u(t,a) = \alpha, \quad u(t,b) = \beta, & t \ge 0 \end{cases}$$

where $g(x)$ is a given function, α and β are given constants. This equation, which is the standard example of Parabolic PDE, is the mathematical model of many important physical process such as the temperature u in a thin wire with the temperature at the ends of the wire held at α and β. The solution $u(t,x)$ then gives the temperature at point x and time t starting from an initial temperature distribution given by $g(x)$.

Difference Equations for the Heat Equation

We first set up a grid in the x,t plane with grid spacings Δx and Δt. Then replace $\dfrac{\partial^2 u}{\partial x^2}$ by central difference in x and replace $\dfrac{\partial u}{\partial t}$ by forward difference in time. Let u_j^m denote the approximate solution at $x_j = a + j\Delta x$ and $t_m = m\Delta t$. Then the finite difference method for the Heat equation is given by

$$\frac{u_j^{m+1} - u_j^m}{\Delta t} = \frac{1}{(\Delta x)^2}(u_{j+1}^m - 2u_j^m + u_{j-1}^m) \qquad j = 1, \ldots, n, \qquad m = 0, 1, \ldots$$

or

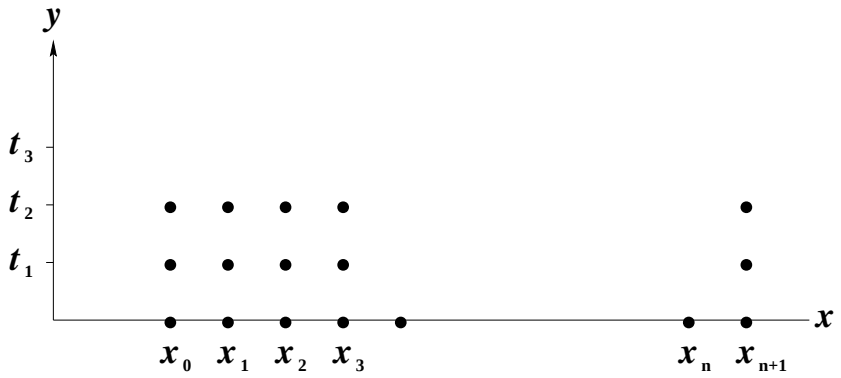
$$u_j^{m+1} = u_j^m + \mu\,(u_{j+1}^m - 2u_j^m + u_{j-1}^m) \tag{3.24}$$

where $\mu = \dfrac{\Delta t}{(\Delta x)^2}$. The boundary conditions are given as the following:

$$u_0^m = \alpha, \qquad u_{n+1}^m = \beta, \qquad m = 0, 1, \ldots$$

and the initial condition is

$$u_j^0 = g(x_j), \qquad j = 1, \ldots, n.$$



Therefore, the formula (3.24) provides a prescription for marching the approximate solution forward one time step after another: first, from the initial values u_j^0 $(j = 0, \ldots, n, n+1)$, the values $u_j^1,\ j = 1, \ldots, n$, are all obtained, and knowing these and boundary values u_0^1 and u_{n+1}^1, we can compute $u_j^2,\ j = 1, \ldots, n$, and so on.

Stability and Convergence

The finite difference formula

$$u_j^{m+1} = u_j^m + \mu\,(u_{j+1}^m - 2u_j^m + u_{j-1}^m), \qquad j = 1, \ldots, n, \qquad m = 0, 1, \ldots$$

can be written in a matrix-vector form

$$U^{m+1} = AU^m$$

where

$$U^m = \begin{bmatrix} u_0^m \\ u_1^m \\ \vdots \\ u_{n+1}^m \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & & & & & \\ \mu & 1-2\mu & \mu & & & \\ & \mu & 1-2\mu & \mu & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu & 1-2\mu & \mu \\ & & & & & 1 \end{bmatrix}$$

It is observed that the matrix A is symmetric. If

$$\mu = \frac{\Delta t}{(\Delta x)^2} \le \frac{1}{2},$$

then

$$|\mu| + |1 - 2\mu| + |\mu| = (1 - 2\mu) + 2\mu = 1.$$

This implies that $||A||_2 = ||A||_\infty = 1$. From $U^{m+1} = AU^m$, we have

$$U^m = A^m U^0\,.$$

Thus, a perturbation ΔU^0 on initial values will cause a change

$$\Delta U^m = A^m \Delta U^0$$

at m^{th} step. Observe that

$$||\Delta U^m|| = ||A^m \Delta U^0|| \leq ||A^m||\,||\Delta U^0|| \leq ||A||^m ||\Delta U^0||\,,$$

we conclude that if $\mu \leq \frac{1}{2}$, i.e.,

$$\Delta t \leq \frac{1}{2}(\Delta x)^2 \tag{3.25}$$

then $||\Delta U^m||_2$ or $||\Delta U^m||_\infty$ will not be enlarged from the initial perturbation for arbitrarily large m. It just means the computation stable.

The relation (3.25) between Δt and Δx is called the Stability Condition.

Theorem 3.3 *The global discretization error*

$$e(\Delta t, \Delta x) = \max\{|u(t_m, x_j) - u_j^m|, \qquad t_m = m\Delta t \qquad x_j = a + j\Delta x\}$$

will converge to zero with the order

$$e = O(\Delta t) + O[(\Delta x)^2]$$

if and only of the method is stable.

The condition (3.25) imposed an increasingly stringent limitation on the time step Δt as the space increment Δx becomes small. This limitation can be removed by the implicit methods discussed next.

Implicit Methods

The previous finite difference method is called explicit because the values of u_j^{m+1} at the $(m+1)^{th}$ time level are obtained by an explicit formula in terms of the values u_j^m, at the previous time level. In contrast, if $\dfrac{\partial u}{\partial t}$ is replaced by a backward difference in time, then the finite difference approximation of heat equation is

$$\frac{u_j^m - u_j^{m-1}}{\Delta t} = \frac{1}{(\Delta x)^2}(u_{j+1}^m - 2u_j^m + u_{j-1}^m), \qquad j = 1, \ldots, n, \qquad m = 1, \ldots$$

In this approximation, the variables u_j^m on the right-hand side are all unknowns thus the equations define the values u_j^m implicitly.

With an implicit method we must solve equations to advance to the next time level. This is the same as implicit methods for ODEs.

Again set $\mu = \dfrac{\Delta t}{(\Delta x)^2}$. Then we can rewrite the finite difference equation as

$$(1 + 2\mu)u_j^m - \mu(u_{j+1}^m + u_{j-1}^m) = u_j^{m-1}, \qquad j = 1, \ldots, n$$

or in matrix-vector form

$$(I + \mu A)U^m = U^{m-1} + b, \qquad m = 1, \ldots$$

where

$$U^m = \begin{bmatrix} u_1^m \\ u_2^m \\ \vdots \\ \vdots \\ u_n^m \end{bmatrix}, \qquad b = \begin{bmatrix} \mu\alpha \\ 0 \\ \vdots \\ 0 \\ \mu\beta \end{bmatrix}, \qquad A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}$$

The implicit method is now carried out by solving the above linear system at each time step to obtain U^m from U^{m-1}. The matrix $(I + \mu A)$ is tridiagonal and also diagonally dominant since $\mu > 0$. We can solve these tridiagonal systems very efficiently. Nevertheless, each time step the implicit method is a little more costly than that of explicit method. However, in payment for this additional cost, we obtain a substantial benefit in the stability properties of the method, which, in many cases, will allow us to use a much larger time step than for explicit method and thus greatly cut the overall computing costs.

Stability and Discretization Error

Let

$$B = (I + \mu A)^{-1}$$

then for initial perturbation ΔU^0 on U^0, the change ΔU^m on U^m is

$$\Delta U^m = B^m \Delta U^0$$

$(I + \mu A)$ is symmetric, so is B. Thus all eigenvalues of B are real. According to Gerschgorin's theorem, all eigenvalues of $(I + \mu A)$ are not less than 1. Hence all eigenvalues of B are not greater than 1. Therefore $||B||_2 = \max_i |\lambda_i^B| \leq 1$. Then, we have

$$\begin{aligned} ||\Delta U^m||_2 &\leq ||B^m||_2 ||\Delta U^0||_2 \leq ||B||_2^m ||\Delta U^0||_2 \\ &\leq ||\Delta U^0||_2 \end{aligned}$$

The inequality shows the stability of the method.

Because there is no restrictions on the relative sizes of Δt and Δx, we say that the method is Unconditionally Stable.

The implicit method is convergent with the discretization error

$$e = O(\Delta t) + O((\Delta x)^2)\,.$$

Thus, we see that although the stability requirements do not impose any restrictions on the relative sizes of Δt and Δx, the accuarcy is affected by the size of Δt.

Crank-Nicolson Method

Making an average of the explicit method and the above implicit method, we obtain the so-called Crank-Nicolson method as follows:

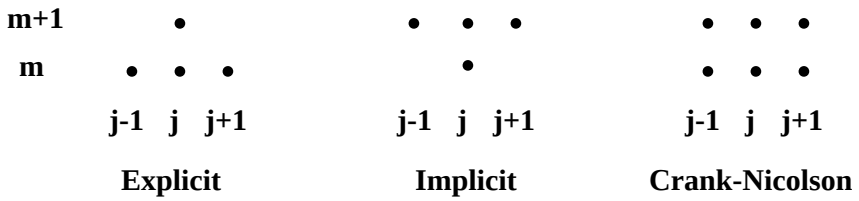
$$u_j^{m+1} - u_j^m = \frac{\Delta t}{2(\Delta x)^2}(u_{j+1}^{m+1} - 2u_j^{m+1} + u_{j-1}^{m+1} + u_{j+1}^m - 2u_j^m + u_{j-1}^m)$$

or in matrix-vector form as

$$(I + \frac{\mu}{2}A)U^{m+1} = (I - \frac{\mu}{2}A)U^m + b, \qquad m = 0, 1, \ldots$$

where U^m, b and A are the same as above.

The Crank-Nicolson method is also implicit. Its advantage is not only unconditionally stable, but it is second-order accurate in time as well as in space. These properties have made it one of the most often used methods for parabolic equations



Stencils for the Methods

3.4.2 Two Dimensional Problems

In the previous section, we considered PDEs in two independent variables: time and one space variable. Since physical phenomena occur in a three-dimensional world, mathematical models in only one space variable are usually considerable simplifications of the actual physical situation although, in many cases, they are sufficient for phenomena that exhibit various symmetries. This section will be concerned with problems in more than one space dimension. For simplicity of exposition, we will only discuss two-dimensional problems.

Finite Difference Approximation for Poisson's Equation

We first consider a time-independent problem,

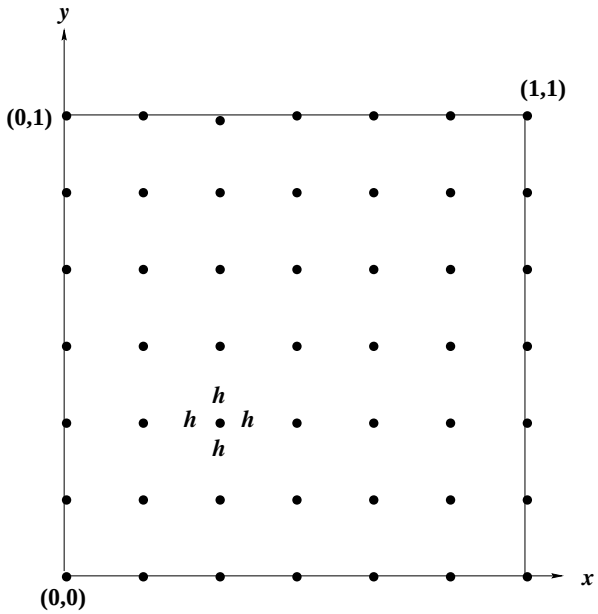
$$u_{xx} + u_{yy} = f(x, y), \qquad (x, y) \in \Omega$$

which is called Poisson's equation. For simplicity, we assume that the domain Ω is the unit square: $0 \leq x,\ y \leq 1$ and that Dirichlet boundary conditions

$$u(x, y) = g(x, y), \qquad (x, y) \text{ on } \partial\,\Omega \text{ — boundary of } \Omega$$

are given, where g is a given function.

We impose a mesh of grid points on the domain Ω with spacing h between the points in both the horizontal and vertical directions. The interior grid points are given by $(x_i, y_j) = (ih, jh), i, j = 1, \ldots, N$ where $h = 1/(N + 1)$.



At an interior grid point (x_i, y_j), we approximate u_{xx} and u_{yy} by the central difference

$$\begin{aligned} u_{xx}(x_i, y_j) &\approx \frac{1}{h^2}[u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)] \\ u_{yy}(x_i, y_j) &\approx \frac{1}{h^2}[u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})] \end{aligned}$$

Putting these approximations into the Poisson's equation, we obtain

$$\frac{1}{h^2}[u(x_{i-1}, y_j) + u(x_{i+1}, y_j) + u(x_i, y_{j-1}) + u(x_i, y_{j+1}) - 4u(x_i, y_j)] \approx f(x_i, y_j)$$

which is an approximate relationship that the exact solution $u(x, y)$ satisfies at any interior grid point.

We now dsefine approximations $u_{i,j}$ to the exact solution $u(x_i, y_j)$, that they satisfy exactly the relationship,

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = -h^2 f_{i,j} \qquad i, j = 1, \ldots, N$$

where we have multiplied the equation by -1.

This is a linear system of N^2 equations with N^2 unknowns: $u_{ij},\ i, j = 1, \ldots, N$. The values of $u_{0,j},\ u_{N+1,j},\ u_{j,0}$ and $u_{j,N+1},\ j = 0, 1, \ldots, N+1$ are known from the boundary condition, that is

$$\begin{aligned} u_{0,j} &= g(0, y_j), \qquad u_{N+1,j} = g(1, y_j) \\ u_{j,0} &= g(x_j, 0), \qquad u_{j,N+1} = g(x_j, 1), \qquad j = 0. \ldots, N+1 \end{aligned}$$

The order $N^2 \times N^2$ coefficient can be written in a Block Tridiagonal Form as

$$A = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{bmatrix}$$

where I denote the $N \times N$ identity matrix and

$$T = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 4 \end{bmatrix}$$

If we define the vectors

$$\begin{aligned} U_i &= (u_{1,i}, u_{2,i}, \ldots, u_{N,i})^T \qquad i = 1, \ldots, N \\ F_i &= (f_{1,i}, f_{2,i}, \ldots, f_{N,i})^T \qquad i = 1, \ldots, N \\ B_1 &= (u_{0,1} + u_{1,0}, u_{2,0}, \ldots, u_{N-1,0}, u_{N,0} + u_{N+1,1})^T \\ B_i &= (u_{0,i}, \underbrace{0, \ldots, 0}_{N-2}, u_{N+1,i})^T, \qquad i = 2, \ldots, N-1 \\ B_N &= (u_{0,N} + u_{1,N+1}, u_{2,N+1}, \ldots, u_{N-1,N+1}, u_{N,N+1} + u_{N+1,N})^T \end{aligned}$$

then we can write the system in the form

$$\begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ \vdots \\ U_N \end{bmatrix} = \begin{bmatrix} B_1 - h^2 F_1 \\ B_2 - h^2 F2 \\ \vdots \\ \vdots \\ B_N - h^2 F_N \end{bmatrix}$$

Stability and Convengence

Assume that there is a perturbation $\delta(x, y)$ on input data $f(x, y)$, the corresponding solution $\tilde{u}_{ij}$ satisfy

$$4\tilde{u}_{ij} - \tilde{u}_{i-1,j} - \tilde{u}_{i+1,j} - \tilde{u}_{i,j-1} - \tilde{u}_{i,j+1} = -h^2(f_{i,j} + \delta_{i,j})$$

It can be shown that for any grid size h there exists a constant C such that

$$\max_{1 \le i \le N,\, 1 \le j \le N} |\tilde{u}_{i,j} - u_{ij}| \le C \max_{1 \le i \le N,\, 1 \le j \le N} |\delta_{i,j}|$$

That means the finite difference approximation is stable. Let

$$e(h) = \max_{1 \le i \le N,\, 1 \le j \le N} |u_{i,j} - u(x_i, y_j)|$$

It can be proved that the discretization error $e(h)$ converges to zero with an order $O(h^2)$.

Heat Equation in Two Dimensions

The mathematical model of the temperature in a flat thin plate is the equation

$$u_t = u_{xx} + u_{yy}, \qquad (x, y) \in \Omega\,.$$

The simplest boundary conditions are when the temperature is prescribed on the boundary of the plate

$$u(t,x,y) = g(x,y), \qquad (x,y) \in \partial\,\Omega$$

where g is a given function. If no heat transfers across the boundary, the boundary condition is

$$\frac{\partial u}{\partial n}(t,x,y) = 0, \qquad (x,y) \in \partial\,\Omega.$$

As with two-point boundary value problems, these two types of boundary condition are called Dirichlet and Neumann respectively.

Again for simplicity in exposition, we will assume that Ω is the unit square and boundary condition is Dirichlet type.

Approximate u_t by forward difference and approximate u_{xx} and u_{yy} by central difference. We obtain the following explicit method

$$u_{i,j}^{m+1} = u_{i,j}^m + \frac{\Delta t}{h^2}(u_{i,j+1}^m + u_{i,j-1}^m + u_{i+1,j}^m + u_{i-1,j}^m - 4u_{i,j}^m)$$

$$i,j = 1,\ldots,N, \qquad m = 0,1,\ldots$$

where $u_{i,j}^m$ denotes the approximate solution at the (t_m, x_i, y_j) with $t_m = m\Delta t,\ x_i = ih,\ y_j = jh$.

This explicit method has the same properties as its one-dimensional counterpart: it is first-order accurate in time and second-order accurate in space; and it is also subject to a stability condition

$$\frac{\Delta t}{h^2} \le \left(\frac{1}{2}\right)^2 = \frac{1}{4}$$

We can attempt to remove this restriction by the use of implicit methods. For example, approximating u_t by backward difference rather than forward difference, we obtain the implicit method

$$u_{i,j}^{m+1} = u_{i,j}^m + \frac{\Delta t}{h^2}(u_{i,j+1}^{m+1} + u_{i,j-1}^{m+1} + u_{i+1,j}^{m+1} + u_{i-1,j}^{m+1} - 4u_{i,j}^{m+1})$$

which is unconditionally stable. However, to carry out this method requires the solution at each time step of the solution of linear equations

$$\left(4 + \frac{h^2}{\Delta t}\right) u_{i,j}^{m+1} - u_{i,j+1}^{m+1} - u_{i,j-1}^{m+1} - u_{i+1,j}^{m+1} - u_{i-1,j}^{m+1} = \frac{h^2}{\Delta t} u_{i,j}^m\,.$$

This system has the same form as the discretization system of Poisson's equation, with the exception that the coefficient of u_{ij}^{m+1} is modified.

The above implicit method has the same convergent order as the explicit method. The Crank-Nicolson method which is second order accurate in time can also be easily extended to this two-dimensional case.

Problem 3.15 The solution of the initial-boundary value problem

$$\begin{aligned} &u_t = u_{xx}, \quad 0 \le x \le 1, \quad t > 0, \\ &u(x,0) = \begin{cases} 2x, & 0 \le x \le 1/2, \\ 2(1-x), & 1/2 \le x \le 1, \end{cases} \\ &u(0,t) = u(1,t) = 0 \end{aligned}$$

is

$$u(x,t) = \frac{8}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k^2} e^{-k^2\pi^2 t} \sin(k\pi/2) \sin k\pi x.$$

Use the explicit scheme (3.24) with $\tau = h^2/10$, $\tau = h^2$ and $\tau = h$ to determine the numerical approximations of $u(x, 0.1)$ for (a) $h = 0.1$, (b) $h = 0.05$ and (c) $h = 0.025$ (plot the solution using Matlab).

Problem 3.16 Consider the problem

$$\begin{aligned} &u_t - a u_{xx} = f(x,t), \\ &u(x,0) = \phi(x), \\ &u(0,t) = u_0(t), \quad u(1,t) = u_1(t). \end{aligned}$$

For $a = 0.125$, $f(x,t) = xt^2$, $\phi(x) = 2x^3 - x$, $u_0(t) = 0$ and $u_1(t) = 1$, take $h = 0.25$, use the explicit scheme similar to (3.24) to find the solution at $(0.5, 0.5)$.

Problem 3.17 Solve Problem 3.15 by using the implicit scheme, take $h = 0.1$, $h = 0.05$, $h = 0.025$, $h = 0.125$, $h = 0.00625$, and $\tau = h/2$, compare the results to the exact solution for $t = 0.1$ (take 5 terms in the summation of the exact solution).

Problem 3.18 Use the five-point scheme which uses

$$u_{i,j-1}, \quad u_{i-1,j}, \quad u_{i,j}, \quad u_{i+1,j}, \quad u_{i,j+1}$$

to approximate the Poisson equation

$$u_{xx} + u_{yy} = 4 \qquad 0 < x < 1, \quad 0 < y < 1$$

with boundary conditions

$$u(x,0) = x^2, \quad u(x,1) = x^2 + 1, \quad u(0,y) = y^2, \quad u(1,y) = 1 + y^2.$$

(a):Write down the five-point scheme for $\Delta x = \Delta y$.

(b):Given $\Delta x = \Delta y = \frac{1}{3}$. Write the resulting system in the form $A\mathbf{u} = \mathbf{b}$.

(c): Find the approximation solutions to $u_{1,1}, u_{2,1}, u_{2,1}$ and $u_{2,2}$.
Hint: $u_{1,1} + u_{2,1} + u_{1,2} + u_{2,2} = 20/9$