Scientific Computing Homework #3

李阳 11935018

May 8, 2020

Problem 1. *Consider the nonlinear equation*

$$f(x) = x^2 - 2 = 0.$$

(a) With $x_0 = 1$ as a starting point, what is the value of x_1 if you use Newton's method for solving this problem?

(b) With $x_0 = 1$ and $x_1 = 2$ as starting points, what is the value of x_2 if you use the secant method for the same problem?

Solution. (a)

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{1-2}{2} = 1.5.$$

(b)

$$x_2 = x_1 - f(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)} = 2 - 2\frac{2-1}{4-(-1)} = 1.6.$$

□

Problem 2. *Newton's method is sometimes used to implement the built-in root function on a computer, with the initial guess supplied by a lookup table.*
What is the Newton iteration for computing the square root of a positive number y (i.e., for solving the equation $f(x) = x^2 - y = 0$, given y)?

Solution.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - y}{2x_n} = \frac{x_n^2 + y}{2x_n}.$$

□

Problem 3. *Express the Newton iteration for solving each of the following systems of nonlinear equations.*

$$x_1^2 + x_1 x_2^3 = 9,$$
$$3x_1^2 x_2 - x_2^3 = 4.$$

Solution. The Jacobian matrix is given by

$$J(x_1, x_2) = \begin{bmatrix} 2x_1 + x_2^3 & 3x_1 x_2^2 \\ 6x_1 x_2 & 3x_1^2 - 3x_2^2. \end{bmatrix}$$

Solve

$$J(x_1^{(n)}, x_2^{(n)}) \begin{bmatrix} s_1^{(n)} \\ s_2^{(n)} \end{bmatrix} = \mathbf{f}(x_1^{(n)}, x_2^{(n)}).$$

Update solution

$$\begin{bmatrix} x_1^{(n+1)} \\ x_2^{(n+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(n)} \\ x_2^{(n)} \end{bmatrix} + \begin{bmatrix} s_1^{(n)} \\ s_2^{(n)} \end{bmatrix}.$$

□

Problem 4. *Suppose you are using the secant method to find a root x^* of a nonlinear equation $f(x) = 0$. Show that if at any iteration it happens to be the case that either $x_k = x^*$ or $x_{k-1} = x^*$ (but not both), then it will also be true that $x_{k+1} = x^*$.*

Solution.

- If $x_k = x^*$, then

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = x^* - f(x^*)\frac{x^* - x_{k-1}}{f(x^*) - f(x_{k-1})} = x^* - 0 = x^*.$$

- If $x_{k-1} = x^*$, then

$$\begin{aligned} x_{k+1} &= x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = x_k - f(x_k)\frac{x_k - x^*}{f(x_k) - f(x^*)} \\ &= x_k - f(x_k)\frac{x_k - x^*}{f(x_k)} = x_k - (x_k - x^*) \\ &= x^*. \end{aligned}$$

□

Problem 5. *Consider the system of equations*

$$\begin{aligned} x_1 - 1 &= 0, \\ x_1 x_2 - 1 &= 0. \end{aligned}$$

For what starting point or points, if any, will Newton's method for solving this system fail? Why?

Solution. The Jacobian matrix is given by

$$J(x_1, x_2) = \begin{bmatrix} 1 & 0 \\ x_2 & x_1 \end{bmatrix}$$

For starting point on the axis $x_2 (x_1 = 0)$, $J(x_1, x_2)$ is singular, and therefore Newton's method will fail. □

Problem 6. *Given the three data points* $(-1, 1), (0, 0), (1, 1)$*, determine the interpolating polynomial of degree two:*

(a) Using the monomial basis

(b) Using the Lagrange basis

(c) Using the Newton basis

Show that the three representations give the same polynomial.

Solution. (a) Solving the following system of linear equations

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

yields $x_1 = 0, x_2 = 0, x_3 = 1$, so that the interpolating polynomial is

$$p_2(t) = t^2.$$

(b) Apply Lagrange interpolation polynomial, and we have

$$\begin{aligned} p_2(t) &= 1 \cdot \frac{(t-0)(t-1)}{(-1-0)(-1-1)} + 0 \cdot \frac{(t+1)(t-1)}{(0+1)(0-1)} + 1 \cdot \frac{(t+1)(t-0)}{(1+1)(1-0)} \\ &= \frac{t(t-1)}{2} + \frac{t(t+1)}{2} \\ &= t^2. \end{aligned}$$

(c) (a) From the table of divided differences

t	y		
-1	1		
0	0	-1	
1	1	1	1

one obtains by Newton's formula

$$p_2(t) = 1 - (t+1) + (t+1)t = t^2.$$

We can see that the above three representations give the same polynomial

$$p_2(t) = t^2.$$

□

Problem 7. *Write a formal algorithm for evaluating a polynomial at a given argument using Horner's nested evaluation scheme*

(a) For a polynomial expressed in terms of the monomial basis

(b) For a polynomial expressed in Newton form

Solution. (a) The pseudocode is given as follows.

Algorithm 1 Evaluate a polynomial $p(t, \mathbf{a}) = \sum_{i=0}^{n} a_i t^i$

```
Input :  $\mathbf{a} = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \end{bmatrix}^T$ and $t$
Output :  $y = p(t, \mathbf{a})$
$y \leftarrow a_n$
for $i = n-1 : -1 : 0$ do
    $y \leftarrow tp + a_i$
end for
```

(b) The pseudocode is given as follows.

Algorithm 2 Evaluate $p(t, \mathbf{a}, \mathbf{t}) = \sum_{i=0}^{n} a_i \prod_{j=0}^{i-1} (t - t_j)$

```
Input :  $\mathbf{a} = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \end{bmatrix}^T$, $\mathbf{t} = \begin{bmatrix} t_0 & t_1 & \cdots & t_{n-1} \end{bmatrix}$ and $t$
Output :  $y = p(t, \mathbf{a}, \mathbf{t})$
$y \leftarrow a_n$
for $i = n-1 : -1 : 0$ do
    $y \leftarrow (t - t_i)p + a_i$
end for
```

□

Problem 8. *Use Lagrange interpolation to derive the formulas given in Section 5.5.5 for inverse quadratic interpolation.*

Solution. The interpolation condition is

$$p_2(f_a) = a, \quad p_2(f_b) = b, \quad p_2(f_c) = c.$$

Applying the Lagrange interpolation polynomial yields

$$p_2(y) = a\frac{(y - f_b)(y - f_c)}{(f_a - f_b)(f_a - f_c)} + b\frac{(y - f_a)(y - f_c)}{(f_b - f_a)(f_b - f_c)} + c\frac{(y - f_a)(t - f_b)}{(f_c - f_a)(f_c - f_b)},$$

evaluating $p_2(y)$ at $y = 0$ gives

$$\begin{aligned} p_2(0) &= a\frac{f_b f_c}{(f_a - f_b)(f_a - f_c)} + b\frac{f_a f_c}{(f_b - f_a)(f_b - f_c)} + c\frac{f_a f_b}{(f_c - f_a)(f_c - f_b)} \\ &= b + \frac{v(w(u-w)(c-b) - (1-u)(b-a))}{(w-1)(u-1)(v-1)}, \end{aligned}$$

where

$$u = \frac{f_b}{f_c}, \quad v = \frac{f_b}{f_a}, \quad w = \frac{f_a}{f_c}.$$

□

Problem 9. *Prove that the formula using divided differences given in Section 7.3.3,*

$$x_j = f[t_1, t_2, \ldots, t_j],$$

indeed gives the coefficient of the jth basis function in the Newton polynomial interpolant.

Proof. We utilize a mathematical induction on j.

- For $j = 1$, the interpolating polynomial is
$$p_1(t) = f(t_1),$$
and hence $x_1 = f(t_1) = f[t_1]$.
- Suppose the conclusion is true for all integers less than n, we show that it holds for $n+1$ as well. By our inductive hypothesis, we know that the polynomial interpolating
$$p_n(t_1) = f(t_1), \quad p_n(t_2) = f(t_2), \quad p_n(t_n) = f(t_n)$$
is given by
$$p_n(t) = \sum_{i=1}^{n} x_i \prod_{j=1}^{i-1}(t - t_j) = \sum_{i=1}^{n} f[t_1, \ldots, t_i] \prod_{j=1}^{i-1}(t - t_j).$$
And the polynomial interpolating
$$q_n(t_2) = f(t_2), \quad q_n(t_3) = f(t_3), \quad q_n(t_{n+1}) = f(t_{n+1})$$
is given by
$$q_n(t) = \sum_{i=1}^{n} x_i \prod_{j=2}^{i}(t - t_j) = \sum_{i=1}^{n} f[t_2, \ldots, t_{i+1}] \prod_{j=2}^{i}(t - t_j).$$
Therefore from the uniqueless of the interpolating polynomial, we know that the polynomial for interpolating
$$p_{n+1}(t_1) = f(t_1), \quad p_{n+1}(t_2) = f(t_2), \quad p_{n+1}(t_n) = f(t_n), \quad p_{n+1}(t_{n+1}) = f(t_{n+1})$$
is given by
$$p_{n+1}(t) = \frac{t - t_{n+1}}{t_1 - t_{n+1}} p_n(t) + \frac{t - t_1}{t_{n+1} - t_1} q_n(t)$$
Comparing the coefficient of the highest-order term of the above two polynomials yields
$$x_{n+1} = \frac{f[t_2, t_3, \ldots, t_{n+1}] - f[t_1, t_2, \ldots, t_n]}{t_{n+1} - t_1} = f[t_1, t_2, \ldots, t_{n+1}],$$
where the second equality follows from the definition of divided differences. Therefore we have shown that the conclusion holds for $n+1$, which completes the inductive proof.

□

Problem 10. *Verify the properties of B-splines enumerated in Section 7.4.3.*

Solution. First let's review the definition of B-splines.

Definition. *B-splines are defined recursively by*

$$B_i^{k+1}(t) = \frac{t - t_i}{t_{i+k+1} - t_i} B_i^k(t) + \frac{t_{i+k+2} - t}{t_{i+k+2} - t_{i+1}} B_{i+1}^k(t). \tag{1}$$

The recursion base is the B-spline of degree zero,

$$B_i^0(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We need to verify the following properties of B-splines, which we decompose into several propositions.

Proposition 1. *For $t < t_i$ or $t > t_{i+k+1}$, $B_i^k(t) = 0$.*

Proof. The induction basis clearly holds because of (2). Now suppose the conclusion holds for some k, then for $k+1$,

$$B_i^{k+1}(t) = \frac{t-t_i}{t_{i+k+1}-t_i} B_i^k(t) + \frac{t_{i+k+2}-t}{t_{i+k+2}-t_{i+1}} B_{i+1}^k(t) = 0$$

for $t < t_i$ or $t > t_{i+k+2}$, since by the induction hypothesis,

$$B_i^k(t) = B_{i+1}^k(t) = 0 \text{ for } t < t_i \text{ or } t > t_{i+k+2}.$$

Therefore the conclusion holds for $k+1$ as well, which completes the proof. □

Proposition 2. *For $t_i < t < t_{i+k+1}$, $B_i^k(t) > 0$.*

Proof. The induction basis clearly holds because of (2). Now suppose the conclusion holds for some k, then by the induction hypothesis and Proposition 1, we have

$$B_i^k(t) > 0 \text{ for } t_i < t < t_{i+k+1} \text{ and } B_i^k(t) = 0 \text{ for } t < t_i \text{ or } t > t_{i+k+1}.$$

$$B_{i+1}^k > 0 \text{ for } t_{i+1} < t < t_{i+k+2} \text{ and } B_{i+1}^k = 0 \text{ for } t < t_{i+1} \text{ or } t > t_{i+k+2}.$$

Combining with (1) gives the conclusion for $k+1$, which completes the proof. □

Proposition 3. *For all t, $\sum_{i=-\infty}^{\infty} B_i^k(t) = 1$.*

Proof. The induction basis clearly holds because of (2). Now suppose the conclusion holds for some k, then for $k+1$, we have

$$\begin{aligned}
\sum_{i=-\infty}^{\infty} B_i^{k+1}(t) &= \sum_{i=-\infty}^{\infty} \left(\frac{t-t_i}{t_{i+k+1}-t_i} B_i^k(t) + \frac{t_{i+k+2}-t}{t_{i+k+2}-t_{i+1}} B_{i+1}^k(t) \right) \\
&= \sum_{i=-\infty}^{\infty} \frac{t-t_i}{t_{i+k+1}-t_i} B_i^k(t) + \sum_{i=-\infty}^{\infty} \frac{t_{i+k+2}-t}{t_{i+k+2}-t_{i+1}} B_{i+1}^k(t) \\
&= \sum_{i=-\infty}^{\infty} \frac{t-t_i}{t_{i+k+1}-t_i} B_i^k(t) + \sum_{i=-\infty}^{\infty} \frac{t_{i+k+1}-t}{t_{i+k+1}-t_i} B_i^k(t) \\
&= \sum_{i=-\infty}^{\infty} \left(\frac{t-t_i}{t_{i+k+1}-t_i} + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_i} \right) B_i^k(t) \\
&= \sum_{i=-\infty}^{\infty} B_i^k(t) = 1,
\end{aligned}$$

where the last equality follows from the induction hypothesis. Hence the conclusion holds for $k+1$ as well, which completes the inductive proof. □

Proposition 4. *For $k \geq 1$, B_i^k is $k-1$ times continuously differentiable.*

Proof. We prove the following theorem:

Theorem. *For $k \geq 2$, we have, $\forall t \in \mathbb{R}$,*

$$\frac{\mathrm{d}}{\mathrm{d}t} B_i^k(t) = \frac{k B_i^{k-1}(t)}{t_{i+k}-t_i} - \frac{k B_{i+1}^{k-1}(t)}{t_{i+k+1}-t_{i+1}}. \tag{3}$$

For $k = 1$, (3) holds for all t except at the three knots t_i, t_{i+1}, t_{i+2}, where the derivative of B_i^1 is not defined.

Proof of Theorem. We first show that (3) holds for all t except at the knots t_j. By (1) and (2), we have

$$\forall t \in \mathbb{R} \backslash \{t_i, t_{i+1}, t_{i+2}\}, \quad \frac{\mathrm{d}}{\mathrm{d}t} B_i^1(t) = \frac{1}{t_{i+1}-t_i} B_i^0(t) - \frac{1}{t_{i+2}-t_{i+1}} B_{i+1}^0(t).$$

Hence the induction hypothesis holds. Now suppose (3) holds $\forall t \in \mathbb{R} \backslash \{t_i, \ldots, t_{i+k+1}\}$. Differentiate (1), apply the induction hypothesis (3), and we have

$$\frac{\mathrm{d}}{\mathrm{d}t} B_i^{k+1}(t) = \frac{B_i^k(t)}{t_{i+k+1}-t_i} - \frac{B_{i+1}^k(t)}{t_{i+k+2}-t_{i+1}} + kC(t) \tag{4}$$

where

$$\begin{aligned}
C(t) &= \frac{t-t_i}{t_{i+k+1}-t_i}\left[\frac{B_i^{k-1}(t)}{t_{i+k}-t_i}-\frac{B_{i+1}^{k-1}(t)}{t_{i+k+1}-t_{i+1}}\right]+\frac{t_{i+k+2}-t}{t_{i+k+2}-t_{i+1}}\left[\frac{B_{i+1}^{k-1}(t)}{t_{i+k+1}-t_{i+1}}-\frac{B_{i+2}^{k-1}(t)}{t_{i+k+2}-t_{i+2}}\right]\\
&= \frac{1}{t_{i+k+1}-t_i}\left[\frac{(t-t_i)B_i^{k-1}(t)}{t_{i+k}-t_i}+\frac{(t_{i+k+1}-t)B_{i+1}^{k-1}(t)}{t_{i+k+1}-t_{i+1}}\right]\\
&\quad -\frac{1}{t_{i+k+2}-t_{i+1}}\left[\frac{(t-t_{i+1})B_{i+1}^{k-1}(t)}{t_{i+k+1}-t_{i+1}}+\frac{(t_{i+k+2}-t)B_{i+2}^{k-1}(t)}{t_{i+k+2}-t_{i+2}}\right]\\
&= \frac{B_i^k(t)}{t_{i+k+1}-t_i}-\frac{B_{i+1}^k(t)}{t_{i+k+2}-t_{i+1}},
\end{aligned}$$

where the last step follows from (1). Then (4) can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}B_i^{k+1}(t)=\frac{(k+1)B_i^k(t)}{t_{i+k+1}-t_i}-\frac{(k+1)B_{i+1}^k(t)}{t_{i+k+2}-t_{i+1}},$$

which completes the inductive proof of (3) except at the knots. Since $B_i^1(t)$ is continuous, an easy induction with (1) shows that B_i^k is continuous for all $k \geq 1$. Hence the right-hand side of (3) is continuous for all $k \geq 2$. Therefore, if $k \geq 2$, $\frac{\mathrm{d}}{\mathrm{d}t}B_i^k(t)$ exists for all $t \in \mathbb{R}$. This completes the proof of the theorem. □

The proof follows from the above theorem and a simple induction on k. □

Proposition 5. *The set of functions $\{B_{1-k}^k, \dots, B_{n-1}^k\}$ is linearly independent on the interval $[t_1, t_n]$.*

Proof.

Lemma. *For $k \geq 2$, we have*

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=-\infty}^{\infty} c_i B_i^k(t) = k\sum_{i=-\infty}^{\infty}\left(\frac{c_i-c_{i-1}}{t_{i+k}-t_i}\right)B_i^{k-1}(t). \tag{5}$$

Proof of Lemma. Utilize (3) and sum over $i=-\infty$ to ∞, and we have the desired result. □

Lemma. *The set of B-splines $\{B_j^k, B_{j+1}^k, \dots, B_{j+k}^k\}$ is linearly independent on $[t_{k+j}, t_{k+j+1}]$.*

Proof of Lemma. First consider the case $k=0$. The lemma asserts that $\{B_j^0\}$ is linearly independent on the interval $[t_j, t_{j+1}]$. This is obviously true. For the purposes of an inductive proof, let $k \geq 1$, and assume that the lemma is correct for index $k-1$. On the basis of this assumption, we shall prove the lemma for the index k. Let $S(t)=\sum_{i=0}^k c_{j+i}B_{j+i}^k(t)$, and suppose that $S|_{[t_{k+j},t_{k+j+1}]}=0$. By (5),

$$0 = S'|_{(t_{k+j},t_{k+j+1})} = k\sum_{i=1}^{k}\frac{c_{j+i}-c_{j+i-1}}{t_{j+i+k}-t_{j+i}}B_{j+i}^{k-1}|_{(t_{k+j},t_{k+j+1})}.$$

To arrive at this equation, we used $B_{j+k+1}^{k-1}=0$ and $B_j^{k-1}=0$ on (t_{k+j}, t_{k+j+1}). By applying the induction hypothesis to $\{B_{j+1}^{k-1}, B_{j+2}^{k-1}, \dots, B_{j+k}^{k-1}\}$, we conclude that this set is linearly independent on the interval (t_{k+j}, t_{k+j+1}). Therefore, in (5) all the coefficients must be 0, and thus we have $c_0=c_1=\cdots=c_k$. If this common value is denoted by λ, we have $S(t)=\lambda$ on (t_{k+j}, t_{k+j+1}) by Proposition 3. (Observe that in Proposition 3, the only terms that are nonzero on the interval (t_{k+j}, t_{k+j+1}) are $B_j^k, B_{j+1}^k, \dots, B_{j+k}^k$.) Since it has been assumed that S vanished on (t_{k+j}, t_{k+j+1}), we conclude that $\lambda=0$. □

Let $S(t)=\sum_{i=1-k}^{n-1}c_iB_i^k(t)$, and suppose that $S|_{[t_1,t_n]}=0$. On the interval $[t_1,t_2]$ only $B_{1-k}^k, B_{2-k}^k, \dots, B_0^k$ are nonzero, and therefore

$$0 = S|_{[t_1,t_2]} = \sum_{i=1-k}^{0} c_iB_i^k|_{[t_1,t_2]}. \tag{6}$$

By the above lemma, the set $\{B_{1-k}^k, B_{2-k}^k, \dots, B_0^k\}$ is linearly independent on (t_1,t_2). Hence from (6), we infer that $c_i=0$ when $1-k \leq i \leq 0$. If all the c_i's are 0, we have the desired conclusion. If not all the

c_i's are 0, let j be the first index for which $c_j \neq 0$. By the prior work, $j \geq 1$. Hence $(t_j, t_{j+1}) \subseteq (t_1, t_n)$. For any $t \in (t_j, t_{j+1})$, we obtain the contradiction

$$0 = S(t) = \sum_{i=j}^{n-1} c_i B_i^k(t) = c_j B_j^k(t) \neq 0.$$

Hence, all the c_i's are 0. □

Proposition 6. *The set of functions $\{B_{1-k}^k, \cdots, B_{n-1}^k\}$ spans the set of all splines of degree k having knots t_i.*

Proof. Combining Proposition 5 and the following two lemmas completes the proof.

Lemma. *If $\mathcal{V}$ is a finite-dimensional linear space, then every linearly independent list of vectors in $\mathcal{V}$ with length* $\dim \mathcal{V}$ *is a basis of $\mathcal{V}$.*

Lemma. *Denote*

$$\mathbb{S}_k^{k-1} = \{s : s \in \mathcal{C}^{k-1}[a,b]; \forall i \in [1, n-1], s|_{[t_i, t_{i+1}]} \in \mathbb{P}_k\}.$$

Then $\mathbb{S}_k^{k-1}(t_1, t_2, \ldots, t_n)$ is a linear space with dimension $k + n - 1$.

□

□

Scientific Computing Homework #4

李阳 11935018

May 16, 2020

Problem 1. *(a) What is the degree of Simpson's rule for numerical quadrature?*

(b) What is the degree of an n-point Gaussian quadrature rule?

Solution. (a) 3.

(b) $2n - 1$.

□

Problem 2. *What is the degree of each of the following types of numerical quadrature rules?*

(a) An n-point Newton-Cotes rule, where n is odd

(b) An n-point Newton-Cotes rule, where n is even

(c) An n-point Gaussian rule

(d) What accounts for the difference between the answers to parts a and b?

(e) What accounts for the difference between the answers to parts b and c?

Solution. (a) n.

(b) $n - 1$.

(c) $2n - 1$.

(d) The difference is due to cancellation of positive and negative errors, as illustrated for the midpoint and Simpson's rules in the following figure, which, on the left, shows a linear polynomial and the constant function interpolating it at the midpoint and, on the right, a cubic and the quadratic interpolaing it at the midpoints and endpoints. Integration of the linear polynomial by the midpoint rule yields two congruent triangles of equal area. The inclusion of one of the triangles compensates exactly for the omission of the other. A simiar phenomenon occurs for the cubic polynomial, where the two shaded regions also have equal areas, so that the addition of one compensates for the subtraction of the other. Such cancellation does not occur, however, for an n-point Newton-Cotes rule if n is even.

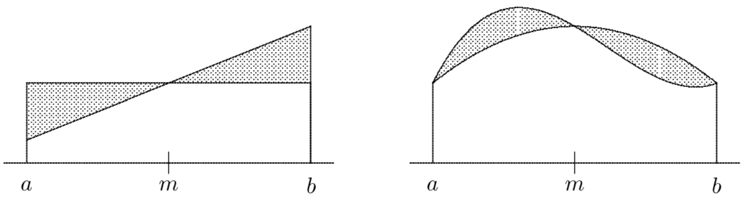

Figure 1: Cancellation of errors in midpoint (left) and Simpson (right) rules.

(e)
- In Newton-Cotes rules, the n nodes are prespecified and the n corresponding weights are then optimally chosen to maximize the degree of the resulting quadrature rules. With only n parameters free to be chosen, the resulting degree is generally $n - 1$.
- In Gaussian rule, the locations of the nodes are also freely chosen, then there are $2n$ free parameters, so that a degree of $2n - 1$ is achievable.

□

Problem 3. *Rank the following types of quadrature rules in order of their degree for the same number of nodes (1 for highest degree, etc.):*

(a) Newton-Cotes

(b) Gaussian

(c) Kronrod

Solution. 1. Gaussian; 2. Kronrod; 3. Newton-Cotes. □

Problem 4. *Why is Monte Carlo* not *a practical method for computing one-dimensional integrals?*

Solution. The error goes to zero as $1/\sqrt{n}$, which means, for example, that to gain an additional decimal digit of accuracy the number of sample points must be increased by a factor of 100. Therefore, for computing one-dimensional integrals, Monte Carlo method is so inefficient, which may require millions of evaluations of the integrand. □

Problem 5. *Explain how a quadrature rule can be used to solve an integral equation numerically. What type of computational problem results?*

Solution. We approximate the integral equation

$$\int_a^b K(s,t)u(t)\,\mathrm{d}t = f(s),$$

by

$$\sum_{j=1}^n w_j K(s_i,t_j)u(t_j) = f(s_i), \quad i = 1,\dots,n,$$

where t_j and w_j $(j = 1,\dots,n)$ are the nodes and weights of a quadrature rule.
Now we can solve the above *system of linear algebratic equations* $A\mathbf{x} = \mathbf{b}$, where $a_{ij} = w_j K(s_i,t_j), b_i = f(s_i)$, and $x_j = u(t_j)$, which can be solved for $\mathbf{x}$ to obtain a discrete sample of approximate values of the solution function u.
As we have seen, the result is solving a system of linear algebraic equations. □

Problem 6. *With an initial value of $y_0 = 1$ at $t_0 = 0$ and a time step of $h = 1$, compute the approximate solution value y_1 at time t_1 for the ODE $y' = -y$ using each of the following two numerical methods. (Your answers should be numbers, not formulas.)*

(a) Euler's method

(b) Backward Euler method

Solution. (a)

$$y_1 = y_0 + hf(t_0,y_0) = y_0 - hy_0 = 1 - 1 = 0.$$

(b)

$$y_1 = y_0 + hf(t_1,y_1) = y_0 - hy_1 \Rightarrow y_1 = \frac{y_0}{1+h} = \frac{1}{1+1} = 0.5.$$

□

Problem 7. *Consider the IVP*

$$y'' = y$$

for $t \geq 0$, with initial values $y(0) = 1$ and $y'(0) = 2$.

(a) Express this second-order ODE as an equivalent system of two first-order ODEs.

(b) What are the corresponding initial conditions for the system of ODEs in part a?

(c) Are solutions of this system stable?

(d) Perform one step of Euler's method for this ODE system using a step size of $h = 0.5$.

(e) Is Euler's method stable for this problem using this step size?

(f) Is the backward Euler method stable for this problem using this step size?

Solution. (a) Define the new unknowns $u_1(t) = y(t)$ and $u_2(t) = y'(t)$, then we have

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \begin{bmatrix} u_2 \\ u_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

(b)

$$\begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

(c) The eigenvalues of the matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

are $1(> 0)$ and -1, thus solutions of this system are unstable.

(d)

$$\mathbf{u}_1 = \mathbf{u}_0 + hA\mathbf{u}_0 = \begin{bmatrix} 1 & 2 \end{bmatrix}^T + 0.5 \begin{bmatrix} 2 & 1 \end{bmatrix}^T = \begin{bmatrix} 2 & 2.5 \end{bmatrix}^T.$$

(e) The eigenvalues of the matrix $I + hA$ are $1.5(> 1)$ and 0.5, therefore, Euler's method is unstable for this problem using this step size.

(f) The formula for the backward Euler method is

$$\mathbf{u}_{n+1} = \mathbf{u}_n + hA\mathbf{u}_{n+1} \Rightarrow \mathbf{u}_{n+1} = (I - hA)^{-1}\mathbf{u}_n,$$

the eigenvalues of the matrix $(I - hA)^{-1}$ are $2(> 1)$ and $2/3$, therefore, backward Euler's method is unstable for this problem using this step size.

□

Problem 8. *Applying the midpoint quadrature rule on the interval* $[t_k, t_{k+1}]$ *leads to the implicit* midpoint method

$$y_{k+1} = y_k + h_k f(t_k + h_k/2, (y_k + y_{k+1})/2)$$

for solving the ODE $y' = f(t, y)$. *Determine the order of accuracy and the stability region of this method.*

Solution. Applying Taylor's theorem yields

$$\begin{aligned} y(t_{k+1}) &= y(t_k) + h_k y'(t_k) + \frac{h_k^2}{2} y''(t_k) + \mathcal{O}(h_k^3); \\ f\left(t_k + \frac{h_k}{2}, \frac{y(t_k) + y(t_{k+1})}{2}\right) &= f(t_k, y(t_k)) + \frac{h_k}{2} f_t(t_k, y(t_k)) + \frac{y(t_{k+1}) - y(t_k)}{2} f_y(t_k, y(t_k)) + \mathcal{O}(h_k^2) \\ &= y'(t_k) + \frac{h_k}{2}\left(f_t(t_k, y(t_k)) + f_y(t_k, y(t_k)) y'(t_k)\right) + \mathcal{O}(h_k^2) \\ &= y'(t_k) + \frac{h_k}{2} y''(t_k) + \mathcal{O}(h_k^2). \end{aligned}$$

Therefore

$$\begin{aligned} y(t_{k+1}) - \left[y(t_k) + h_k f\left(t_k + \frac{h_k}{2}, \frac{y(t_k) + y(t_{k+1})}{2}\right)\right] &= y(t_k) + h_k y'(t_k) + \frac{h_k^2}{2} y''(t_k) + \mathcal{O}(h_k^3) \\ &\quad - \left\{y(t_k) + h_k \left[y'(t_k) + \frac{h_k}{2} y''(t_k) + \mathcal{O}(h_k^2)\right]\right\} \\ &= \mathcal{O}(h_k^3), \end{aligned}$$

which shows that the implicit midpoint method is of order 2.
To determine the stability of the implicit midpoint method, we apply it to the scalar test ODE $y' = \lambda y$, obtaining

$$y_{k+1} = y_k + \frac{\lambda h_k}{2}(y_k + y_{k+1}),$$

which implies that

$$y_k = \left(\frac{1 + h_k\lambda/2}{1 - h_k\lambda/2}\right)^k y_0.$$

Thus, the stability region of the implicit midpoint method is

$$\mathcal{D} = \left\{z \in \mathbb{C} : \left|\frac{1 + z}{1 - z}\right| < 1\right\}.$$

□

Problem 9. *Consider the two-point BVP for the second-order scalar ODE*

$$u'' = u, \quad 0 < t < b,$$

with boundary conditions

$$u(0) = \alpha, \quad u(b) = \beta.$$

(a) Rewrite the problem as a first-order system of ODEs with separated boundary conditions.

(b) Show that the fundamental solution matrix for the resulting linear system of ODEs is given by

$$Y(t) = \begin{bmatrix} \cosh(t) & \sinh(t) \\ \sinh(t) & \cosh(t) \end{bmatrix}.$$

(c) Are the solutions to this ODE stable?

(d) Determine the matrix $Q \equiv B_0 Y(0) + B_b Y(b)$ for this problem.

(e) Determine the rescaled solution matrix $\Phi(t) = Y(t)Q^{-1}$.

(f) What can you say about the conditioning of Q, the norm of $\Phi(t)$, and the stability of solutions to this BVP as the right endpoint b grows?

Solution. (a) Define the new unknowns $y_1(t) = u(t)$ and $y_2(t) = u'(t)$, then we have

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

with separated linear boundary conditions

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

(b) Solving $\mathbf{y}' = A\mathbf{y}$ with initial condition $\mathbf{y}(0) = \mathbf{e}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, we obtain $\mathbf{y}(t) = \begin{bmatrix} \cosh(t) & \sinh(t) \end{bmatrix}^T$, with $\mathbf{y}(0) = \mathbf{e}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, $\mathbf{y}(t) = \begin{bmatrix} \sinh(t) & \cosh(t) \end{bmatrix}^T$. Therefore the fundamental solution matrix is

$$Y(t) = \begin{bmatrix} \cosh(t) & \sinh(t) \\ \sinh(t) & \cosh(t) \end{bmatrix}.$$

(c) The solutions to this ODE are stable, since growth in the solution is limited by the boundary conditions.

(d)

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cosh(b) & \sinh(b) \\ -\sinh(b) & \cosh(b) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \cosh(b) & \sinh(b) \end{bmatrix}.$$

(e)

$$\begin{aligned} \Phi(t) = Y(t)Q^{-1} &= \begin{bmatrix} \cosh(t) & \sinh(t) \\ \sinh(t) & \cosh(t) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\cosh(b)}{\sinh(b)} & \frac{1}{\sinh(b)} \end{bmatrix} \\ &= \frac{1}{\sinh(b)} \begin{bmatrix} \sinh(b-t) & \sinh(t) \\ -\cosh(b-t) & \cosh(t) \end{bmatrix} \end{aligned}$$

(f) As b grows, the condition number of Q and the norm of $\Phi(t)$ grow as well, and the stability of solutions to this BVP decreases.

□

Problem 10. *Consider the two-point BVP*

$$u'' = u^3 + t, \quad a < t < b,$$

with boundary conditions

$$u(a) = \alpha, \quad u(b) = \beta.$$

To use the shooting method to solve this problem, one needs a starting guess for the initial slope $u'(a)$. One way to obtain such a starting guess for the initial slope is, in effect, to do a "preliminary shooting" in which we take a single step of Euler's method with $h = b - a$.

(a) Using this approach, write out the resulting algebraic equation for the initial slope.

(b) What starting value for the inital slope results from this approach?

Solution. (a)

$$u(b) = u(a) + hu'(a) \Rightarrow hu'(a) = u(b) - u(a).$$

(b)

$$u'(a) = \frac{u(b) - u(a)}{h} = \frac{\beta - \alpha}{b - a}.$$

□