Section 4

Boundary Value Problems for ODEs

BVP for ODE

We study numerical solution for boundary value problem (BVP).

If the BVP involves first-order ODE, then

$$y'(x) = f(x, y(x)), \quad a \le x \le b, \quad y(a) = \alpha.$$

This reduces to an initial value problem we learned before.

So we start by considering second-order ODE:

$$\begin{cases} y''(x) = f(x, y(x), y'(x)), \quad a \le x \le b \\ y(a) = \alpha, \ y(b) = \beta \end{cases}$$

Existence of solutions

Consider the BVP with second-order ODE:

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

Theorem (Existence and uniqueness of solution)

Let $D = [a, b] \times \mathbb{R} \times \mathbb{R}$. Suppose $f(x, y, y')$ satisfies:

1. *f is continuous on D,*
2. *$\frac{\partial f}{\partial y} > 0$ in D,*
3. *$\exists M > 0$ such that $|\frac{\partial f}{\partial y'}| \le M$ in D.*

Then the BVP has unique solution.

Existence of solutions

Example (Existence and uniqueness of solution)

Show that the BVP below has unique solution:

$$\begin{cases} y''(x) = -e^{-xy} + \sin(y'), \quad 1 \le x \le 2 \\ y(a) = 0, \; y(b) = 0 \end{cases}$$

Solution: We have $f(x, y, y') = -e^{-xy} - \sin(y')$. It is obvious that f is continuous. Moreover $\partial_y f = xe^{-xy} > 0$, and $|\partial_{y'} f| = |-\cos(y')| \le 1$. So the BVP has unique solution by the theorem above.

BVP with linear ODE

Now we first consider a linear second-order ODE:

$$\begin{cases} y'' = p(x)y' + q(x)y + r(x), \quad a \le x \le b \\ y(a) = \alpha, \; y(b) = \beta \end{cases}$$

where $p, q, r : [a, b] \to \mathbb{R}$ are given functions.

Corollary

If p, q, r are continuous on $[a, b]$, $q > 0$ for all x, then the BVP with linear ODE above has a unique solution.

Proof.

Set $f = py' + qy + r$. Note that p is bounded since it is continuous on $[a, b]$. Hence the theorem (check the 3 conditions) above applies. $\square$

Linear shooting method

Now we consider how to solve BVP with linear ODE:

$$\begin{cases} y'' = py' + qy + r, \quad a \le x \le b \\ y(a) = \alpha, \ y(b) = \beta \end{cases}$$

We consider two associated initial value problems:

$$\begin{cases} y_1'' = py_1' + qy_1 + r, \quad a \le x \le b \\ y_1(a) = \alpha, \ y_1'(a) = 0 \end{cases}$$

$$\begin{cases} y_2'' = py_2' + qy_2, \quad a \le x \le b \\ y_2(a) = 0, \ y_2'(a) = 1 \end{cases}$$

Linear shooting method

Suppose the solution y to the BVP can be written as $y = y_1 + cy_2$ for some constant c (to be determined soon), where y_1, y_2 are the solutions to the two IVPs. Then y satisfies the ODE:

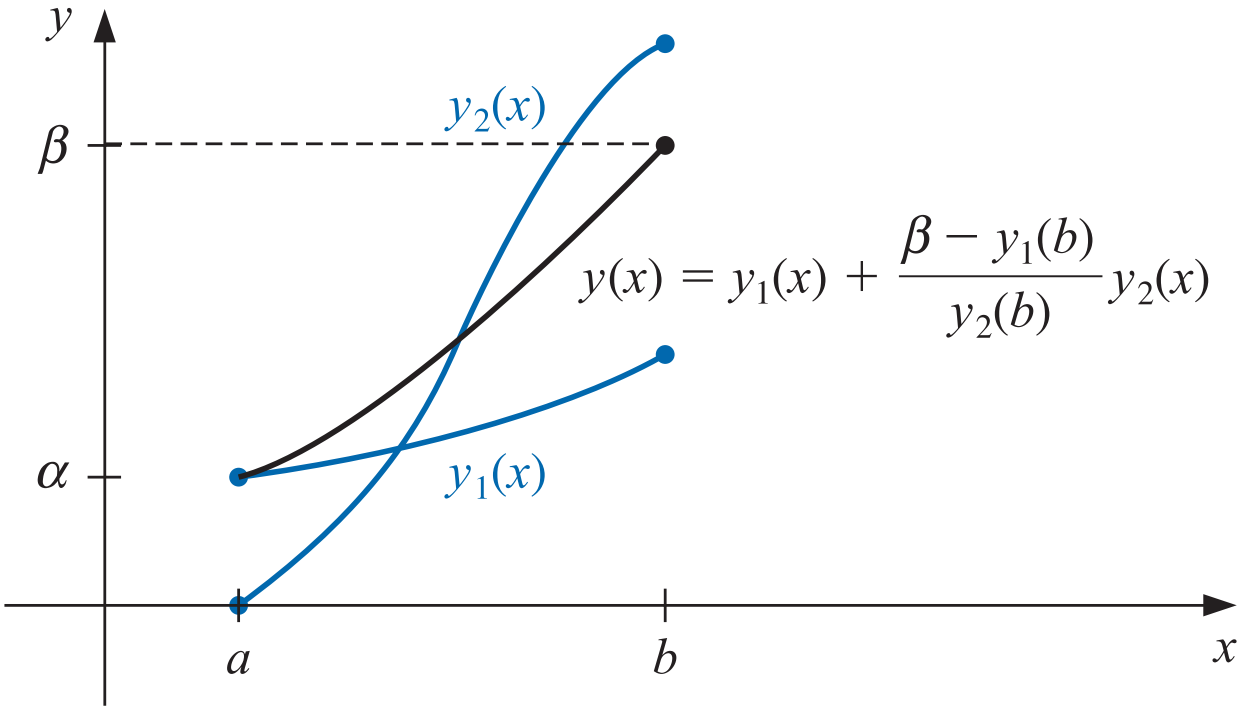
$$
\begin{aligned}
y'' &= (y_1 + cy_2)'' = y_1'' + cy_2'' \\
&= (py_1' + qy_1 + r) + c(py_2' + qy_2) \\
&= p(y_1 + cy_2)' + q(y_1 + cy_2) + r \\
&= py' + qy + r
\end{aligned}
$$

To make y satisfy the boundary conditions, we need c such that

$$
\begin{aligned}
y(a) &= y_1(a) + cy_2(a) = y_1(a) = \alpha \\
y(b) &= y_1(b) + cy_2(b) = \beta
\end{aligned}
$$

So we just need to set $c = \frac{\beta - y_1(b)}{y_2(b)}$.

Linear shooting method



Here y_1, y_2 are two shot trajectories based on their initial height and angle. Their linear combination $y_1 + \frac{\beta - y_1(b)}{y_2(b)} y_2$ is the solution y.

Linear shooting method

Steps of the linear shooting method:

1. Partition $[a, b]$ into N equal subintervals.
2. Solve y_1 and y_2 from their own IVPs (e.g., using RK4) $(u_1 = y_1, u_2 = y_1', v_1 = y_2, v_2 = y_2')$, and get $\{u_{1,i}, v_{1,i} : 0 \le i \le N\}$
3. Set $c = (\beta - u_{1,N})/v_{1,N}$, and set $w_{1,i} = u_{1,i} + cv_{1,i}$ for $0 \le i \le N$.

Linear shooting method

Example (Linear shooting method)

Solve the BVP with $N = 10$.

$$\begin{cases} y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \quad 1 \le x \le 2 \\ y(1) = 1, \; y(2) = 2 \end{cases}$$

Solution: Partition $[1, 2]$ into $N = 10$ subintervals, and solve

$$\begin{cases} y_1'' = -\frac{2}{x}y_1' + \frac{2}{x^2}y_1 + \frac{\sin(\ln x)}{x^2}, \quad 1 \le x \le 2 \\ y_1(1) = 1, \; y_1'(1) = 0 \end{cases}$$

$$\begin{cases} y_2'' = -\frac{2}{x}y_2' + \frac{2}{x^2}y_2, \quad 1 \le x \le 2 \\ y_2(1) = 0, \; y_2'(1) = 1 \end{cases}$$

Then set $w_i = u_{1,i} + \frac{2-u_{1,N}}{v_{1,N}}v_{1,i}$ for $i = 0, \dots, 10$.

Linear shooting method

Numerical result:

x_i	$u_{1,i} \approx y_1(x_i)$	$v_{1,i} \approx y_2(x_i)$	$w_i \approx y(x_i)$	$y(x_i)$	$\lvert y(x_i) - w_i \rvert$
1.0	1.00000000	0.00000000	1.00000000	1.00000000	
1.1	1.00896058	0.09117986	1.09262917	1.09262930	1.43×10^{-7}
1.2	1.03245472	0.16851175	1.18708471	1.18708484	1.34×10^{-7}
1.3	1.06674375	0.23608704	1.28338227	1.28338236	9.78×10^{-8}
1.4	1.10928795	0.29659067	1.38144589	1.38144595	6.02×10^{-8}
1.5	1.15830000	0.35184379	1.48115939	1.48115942	3.06×10^{-8}
1.6	1.21248372	0.40311695	1.58239245	1.58239246	1.08×10^{-8}
1.7	1.27087454	0.45131840	1.68501396	1.68501396	5.43×10^{-10}
1.8	1.33273851	0.49711137	1.78889854	1.78889853	5.05×10^{-9}
1.9	1.39750618	0.54098928	1.89392951	1.89392951	4.41×10^{-9}
2.0	1.46472815	0.58332538	2.00000000	2.00000000	

This accurate result is due to $O(h^4)$ of RK4 used for the two IVPs.

Round-off error in linear shooting method

If $y_1(x)$ grows too fast such that $y_1(b) \gg \beta$, then

$$\frac{\beta - y_1(b)}{y_2(b)} \approx -\frac{y_1(b)}{y_2(b)}$$

which is prone to round-off error.

In this case, we can solve the two IVPs backward in x:

$$\begin{cases} y_1'' = py_1' + qy_1 + r, & a \le x \le b \\ y_1(b) = \beta, \ y_1'(b) = 0 \end{cases}$$

$$\begin{cases} y_2'' = py_2' + qy_2, & a \le x \le b \\ y_2(b) = 0, \ y_2'(b) = 1 \end{cases}$$

and set $y(x) = y_1(x) + \frac{\alpha - y_1(a)}{y_2(a)} y_2(x)$ for $a \le x \le b$

Nonlinear shooting method

Consider the BVP with nonlinear ODE (f is a nonlinear function):

$$\begin{cases} y'' = f(x, y, y'), \quad a \le x \le b \\ y(a) = \alpha, \ y(b) = \beta \end{cases}$$

Suppose we try to solve the IVP with some given t:

$$\begin{cases} y'' = f(x, y, y'), \quad a \le x \le b \\ y(a) = \alpha, \ y'(a) = t \end{cases}$$

and obtain solution $y(x, t)$ (since the solution depends on t) for $a \le x \le b$.

Then we hope to find t such that $y(b, t) = \beta$.

Secant method for nonlinear shooting

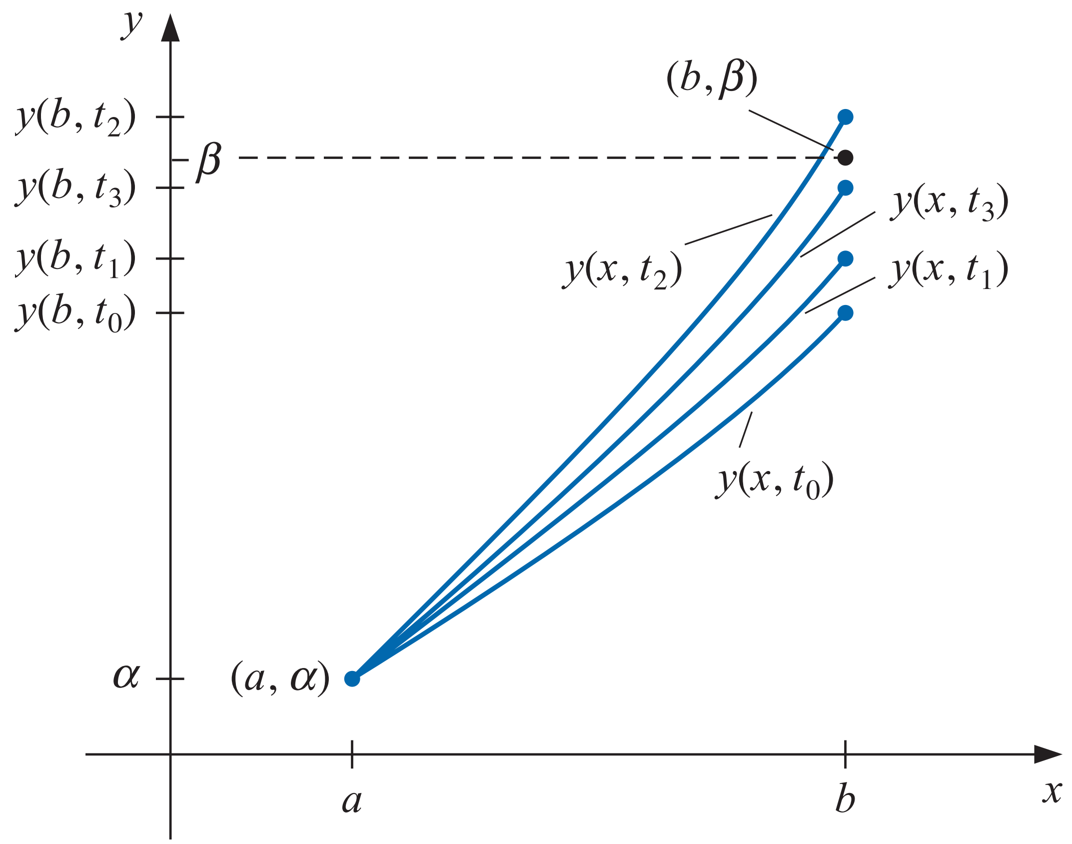
Suppose we have two initials t_0, t_1, then we use the secant method to solve $y(b,t) - \beta = 0$ by iterating

$$t_k = t_{k-1} - \frac{(y(b,t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{y(b,t_{k-1}) - y(b,t_{k-2})}$$

For each k, we need to compute $y(b,t_k)$ by solving the IVP:

$$\begin{cases} y'' = f(x,y,y'), \quad a \le x \le b \\ y(a) = \alpha, \; y'(a) = t_k \end{cases}$$

Nonlinear shooting method



Here $y(x, t_k)$ is "shooting" at an angle (with slope t_k) and try to "hit" β at $x = b$.

Newton's method for nonlinear shooting

We can also consider Newton's method to $y(b,t) - \beta = 0$ for fewer iterations:

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{\partial_t y(b, t_{k-1})}$$

However, we need to know $\partial_t y(b,t)$ ...

We denote the solution of IVP below by $y(x,t)$:

$$\begin{cases} y''(x,t) = f(x, y(x,t), y'(x,t)), & a \le x \le b \\ y(a,t) = \alpha, \;\; y'(a,t) = t \end{cases}$$

where $y' = \partial_x y$ and $y'' = \partial_x^2 y$ (i.e., the $'$ is on x).

Newton's method for nonlinear shooting

Taking partial derivatives with respect to t above yields:

$$\begin{cases} \partial_t y'' = \partial_y f \cdot \partial_t y + \partial_{y'} f \cdot \partial_t y', \quad a \le x \le b \\ \partial_t y(a,t) = 0, \ \partial_t y'(a,t) = 1 \end{cases}$$

Denote $z(x,t) = \partial_t y(x,t)$. Suppose ∂_x and ∂_t can exchange, then

$$\begin{cases} z''(x,t) = \partial_y f \cdot z(x,t) + \partial_{y'} f \cdot z'(x,t), \quad a \le x \le b \\ z(a,t) = 0, \ z'(a,t) = 1 \end{cases}$$

and set $\partial_t y(b,t) = z(b,t)$.

Newton's method for nonlinear shooting

Steps of Newton's method for nonlinear shooting:

1. Initialize t_0 (e.g. $t_0 = \frac{\beta-\alpha}{b-a}$). Set $k = 1$.
2. For $t = t_{k-1}$, solve $y(x,t)$ and $z(x,t)$ from

$$\begin{cases} y''(x,t) = f(x, y(x,t), y'(x,t)), & a \le x \le b \\ y(a,t) = \alpha, \ y'(a,t) = t \end{cases}$$

$$\begin{cases} z''(x,t) = \partial_y f \cdot z(x,t) + \partial_{y'} f \cdot z'(x,t), & a \le x \le b \\ z(a,t) = 0, \ z'(a,t) = 1 \end{cases}$$

   and set $t_k = t_{k-1} - \frac{y(b,t_{k-1})-\beta}{z(b,t_{k-1})}$.
3. Set $k \leftarrow k+1$ and go to Step 2.

Newton's method for nonlinear shooting

Example (Newton's method for nonlinear BVP)

Solve the BVP with nonlinear ODE using Newton's method with $N = 20$ for maximal of 10 iterations or $|w_N(t_k) - y(3)| \le 10^{-5}$:

$$\begin{cases} y'' = \frac{1}{8}\left(32 + 2x^3 - yy'\right), \quad 1 \le x \le 3 \\ y(1) = 17, \ y(3) = \frac{43}{3} \end{cases}$$

Solution: Note that $\partial_y f = -\frac{1}{8}y'$ and $\partial_{y'} f = -\frac{1}{8}y$. For every t, the two IVPs are (note z depends on y but not vice versa):

$$\begin{cases} y'' = \frac{1}{8}\left(32 + 2x^3 - yy'\right), \quad 1 \le x \le 3 \\ y(1) = 17, \ y'(1) = t \end{cases}$$

$$\begin{cases} z'' = -\frac{1}{8}(y'z + yz'), \quad 1 \le x \le 3 \\ z(1) = 0, \ z'(1) = 1 \end{cases}$$

Nonlinear shooting using Newton's method

x_i	$w_{1,i}$	$y(x_i)$	$\lvert w_{1,i} - y(x_i)\rvert$
1.0	17.000000	17.000000	
1.1	15.755495	15.755455	4.06×10^{-5}
1.2	14.773389	14.773333	5.60×10^{-5}
1.3	13.997752	13.997692	5.94×10^{-5}
1.4	13.388629	13.388571	5.71×10^{-5}
1.5	12.916719	12.916667	5.23×10^{-5}
1.6	12.560046	12.560000	4.64×10^{-5}
1.7	12.301805	12.301765	4.02×10^{-5}
1.8	12.128923	12.128889	3.14×10^{-5}
1.9	12.031081	12.031053	2.84×10^{-5}
2.0	12.000023	12.000000	2.32×10^{-5}
2.1	12.029066	12.029048	1.84×10^{-5}
2.2	12.112741	12.112727	1.40×10^{-5}
2.3	12.246532	12.246522	1.01×10^{-5}
2.4	12.426673	12.426667	6.68×10^{-6}
2.5	12.650004	12.650000	3.61×10^{-6}
2.6	12.913847	12.913845	9.17×10^{-7}
2.7	13.215924	13.215926	1.43×10^{-6}
2.8	13.554282	13.554286	3.46×10^{-6}
2.9	13.927236	13.927241	5.21×10^{-6}
3.0	14.333327	14.333333	6.69×10^{-6}

Netwon's method requires solving two IVPs in each iteration, but converges much faster than secant method. Still sensitive to round-off errors if y or z increases rapidly.

Finite-difference method for linear problems

Idea: Partition $[a, b]$ into $N+1$ subintervals with nodes $a = x_0 < \cdots < x_{N+1} = b$ and step size $h = \frac{b-a}{N+1}$. Then approximate y', y'' by finite differences, and solve $w_i = y(x_i)$ for $0 \le i \le N+1$.

Recall the centered-difference approximation of $y'(x_i)$:

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(\eta_i^+)$$
$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(\eta_i^-)$$

where $\eta_i^{\pm}$ is between x_i and $x_{i\pm1}$. Then subtracting the two above:

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{6}y'''(\eta_i)$$

for some $\eta_i \in (x_{i-1}, x_{i+1})$ due to IVT and $y \in C^3$.

Finite-difference method for linear problems

Similarly, we have the centered-difference approximation of $y''(x_i)$:

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^+)$$
$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^-)$$

where $\xi_i^{\pm}$ is between x_i and $x_{i\pm1}$. Then adding the two above:

$$y''(x_i) = \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} - \frac{h^2}{12}y^{(4)}(\xi_i)$$

for some $\xi_i \in (x_{i-1}, x_{i+1})$ due to IVT and $y \in C^4$.

Finite-difference method for linear problems

Plugging the two identities about $y'(x_i)$ and $y''(x_i)$ above into $y'' = py' + qy + r$:

$$\begin{aligned}\frac{y\left(x_{i+1}\right)-2y\left(x_i\right)+y\left(x_{i-1}\right)}{h^2}=&p\left(x_i\right)\left[\frac{y\left(x_{i+1}\right)-y\left(x_{i-1}\right)}{2h}\right]+q\left(x_i\right)y\left(x_i\right)\\&+r\left(x_i\right)-\frac{h^2}{12}\left[2p\left(x_i\right)y'''\left(\eta_i\right)-y^{(4)}\left(\xi_i\right)\right]\end{aligned}$$

which has truncation error $O(h^2)$.

Now we approximate $y(x_i)$ by w_i for $0 \le i \le N+1$. Note that $w_0 = y(a) = \alpha$ and $w_{N+1} = y(b) = \beta$, and for $i = 1, \dots, N$:

$$\left(\frac{-w_{i+1}+2w_i-w_{i-1}}{h^2}\right)+p\left(x_i\right)\left(\frac{w_{i+1}-w_{i-1}}{2h}\right)+q\left(x_i\right)w_i=-r\left(x_i\right)$$

Finite-difference method for linear problems

The equation above can be rearranged into

$$-\left(1+\frac{h}{2}p\left(x_i\right)\right)w_{i-1}+\left(2+h^2q\left(x_i\right)\right)w_i-\left(1-\frac{h}{2}p\left(x_i\right)\right)w_{i+1}=-h^2r\left(x_i\right)$$

This is a linear system $Aw = b$ where $w = (w_1, \dots, w_N)^\top$, A is tridiagonal, and b is known.

Theorem

Suppose that p, q, r are continuous on $[a, b]$ and $q \geq 0$, then the tridiagonal linear system $Aw = b$ has a unique solution provided that $h < 2/L$ where $L = \max_{a \leq x \leq b} |p(x)|$.

Finite-difference method for linear problems

Example (Finite-difference method for linear problems)

Solve the BVP below using finite difference method with $N = 9$:

$$\begin{cases} y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, & 1 \le x \le 2 \\ y(1) = 1, \; y(2) = 2 \end{cases}$$

Solution: Note that $p(x) = -2/x$, $q(x) = 2/x^2$, and $r(x) = \sin(\ln x)/x^2$. Step size $h = (b-a)/(N+1) = 0.1$.

Finite-difference method for linear problems

x_i	w_i	$y(x_i)$	$\lvert w_i - y(x_i) \rvert$
1.0	1.00000000	1.00000000	
1.1	1.09260052	1.09262930	2.88×10^{-5}
1.2	1.18704313	1.18708484	4.17×10^{-5}
1.3	1.28333687	1.28338236	4.55×10^{-5}
1.4	1.38140205	1.38144595	4.39×10^{-5}
1.5	1.48112026	1.48115942	3.92×10^{-5}
1.6	1.58235990	1.58239246	3.26×10^{-5}
1.7	1.68498902	1.68501396	2.49×10^{-5}
1.8	1.78888175	1.78889853	1.68×10^{-5}
1.9	1.89392110	1.89392951	8.41×10^{-6}
2.0	2.00000000	2.00000000	

The error is $O(h^2)$, which is worse than the linear shooting method.

Finite-difference method for linear problems

We can improve the error order by Richardson's extrapolation since the truncation errors are in even orders of h.

Consider the same example above, we use step sizes $h = 0.1$, 0.05, and 0.025 to compute $w(h = 0.1)$, $w(h = 0.05)$ and $w(h = 0.025)$, and compute

$$\begin{aligned}
\mathrm{Ext}_{1i} &= \frac{4w_i(h = 0.05) - w_i(h = 0.1)}{3} \\
\mathrm{Ext}_{2i} &= \frac{4w_i(h = 0.025) - w_i(h = 0.05)}{3} \\
\mathrm{Ext}_{3i} &= \frac{16\mathrm{Ext}_{2i} - \mathrm{Ext}_{1i}}{15}
\end{aligned}$$

Finite-difference method for linear problems

x_i	$w_i(h = 0.05)$	$w_i(h = 0.025)$	Ext_{1i}	Ext_{2i}	Ext_{3i}
1.0	1.00000000	1.00000000	1.00000000	1.00000000	1.00000000
1.1	1.09262207	1.09262749	1.09262925	1.09262930	1.09262930
1.2	1.18707436	1.18708222	1.18708477	1.18708484	1.18708484
1.3	1.28337094	1.28337950	1.28338230	1.28338236	1.28338236
1.4	1.38143493	1.38144319	1.38144589	1.38144595	1.38144595
1.5	1.48114959	1.48115696	1.48115937	1.48115941	1.48115942
1.6	1.58238429	1.58239042	1.58239242	1.58239246	1.58239246
1.7	1.68500770	1.68501240	1.68501393	1.68501396	1.68501396
1.8	1.78889432	1.78889748	1.78889852	1.78889853	1.78889853
1.9	1.89392740	1.89392898	1.89392950	1.89392951	1.89392951
2.0	2.00000000	2.00000000	2.00000000	2.00000000	2.00000000

The error reduces to 6.3×10^{-11} which significantly improves the case with $h = 0.1$ (about 10^{-5}).

Finite-difference method for nonlinear problems

Consider the BVP with nonlinear ODE:

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

Theorem

Let $D = [a, b] \times \mathbb{R} \times \mathbb{R}$. If f satisfies the following conditions:

1. *f is continuous on D,*
2. *$\exists \delta > 0$ such that $\partial_y f(x, y, y') \ge \delta$ on D,*
3. *$\exists L > 0$ such that $|\partial_y f|, |\partial_{y'} f| \le L$ on D.*

Then the BVP has a unique solution.

Finite-difference method for nonlinear problems

We apply the same partition of $[a, b]$ into $N+1$ subintervals and centered-difference approximations for $y'(x_i)$ and $y''(x_i)$: $w_0 = \alpha$, $w_{N+1} = \beta$, and for $i = 1, \dots, N$

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

This is a system of N nonlinear equations of $(w_1, \dots, w_N)$:

$$\begin{aligned}
2w_1 - w_2 + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) - \alpha &= 0 \\
-w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) &= 0 \\
&\vdots \\
-w_{N-2} + 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right) &= 0 \\
-w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta &= 0
\end{aligned}$$

Finite-difference method for nonlinear problems

We can write the system as $F(w) = 0$ (note that $F : \mathbb{R}^N \to \mathbb{R}^N$).

To solve this system, we can apply the Newton's method:

$$w^{(k)} = w^{(k-1)} - J(w^{(k-1)})^{-1} F(w^{(k-1)})$$

starting from some initial value $w^{(0)}$. Here $J(w) \in \mathbb{R}^{N\times N}$ is the Jacobian of $F(w)$.

The key is to solve $v = J(w)^{-1}F(w)$ from $J(w)v = F(w)$ for given w.

Finite-difference method for nonlinear problems

Recall that $F(w) = (F_1(w), \dots, F_N(w))^\top \in \mathbb{R}^N$ where

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

Jacobian $J(w) = [\frac{\partial F_i(w)}{\partial w_j}] \in \mathbb{R}^{N\times N}$ is tridiagonal:

$$\begin{aligned} J(w_1, \dots, w_N)_{ij} &= \frac{\partial F_i(w)}{\partial w_j} \\ &= \begin{cases} -1 + \frac{h}{2} f_{y'}\left(x_i, w_i, \frac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i = j-1 \text{ and } j = 2, \dots, N \\ 2 + h^2 f_y\left(x_i, w_i, \frac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i = j \text{ and } j = 1, \dots, N \\ -1 - \frac{h}{2} f_{y'}\left(x_i, w_i, \frac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i = j+1 \text{ and } j = 1, \dots, N-1 \\ 0, & \text{for } |i-j| > 1 \end{cases} \end{aligned}$$

Finite-difference method for nonlinear problems

Example (Finite-difference method for nonlinear BVP)

Solve the BVP with nonlinear ODE using finite difference method with $h = 0.1$:

$$\begin{cases} y'' = \frac{1}{8}\left(32 + 2x^3 - yy'\right), \quad 1 \le x \le 3 \\ y(1) = 17, \; y(3) = \frac{43}{3} \end{cases}$$

Finite-difference method for nonlinear problems

x_i	w_i	$y(x_i)$	$\lvert w_i - y(x_i) \rvert$
1.0	17.000000	17.000000	
1.1	15.754503	15.755455	9.520×10^{-4}
1.2	14.771740	14.773333	1.594×10^{-3}
1.3	13.995677	13.997692	2.015×10^{-3}
1.4	13.386297	13.388571	2.275×10^{-3}
1.5	12.914252	12.916667	2.414×10^{-3}
1.6	12.557538	12.560000	2.462×10^{-3}
1.7	12.299326	12.301765	2.438×10^{-3}
1.8	12.126529	12.128889	2.360×10^{-3}
1.9	12.028814	12.031053	2.239×10^{-3}
2.0	11.997915	12.000000	2.085×10^{-3}
2.1	12.027142	12.029048	1.905×10^{-3}
2.2	12.111020	12.112727	1.707×10^{-3}
2.3	12.245025	12.246522	1.497×10^{-3}
2.4	12.425388	12.426667	1.278×10^{-3}
2.5	12.648944	12.650000	1.056×10^{-3}
2.6	12.913013	12.913846	8.335×10^{-4}
2.7	13.215312	13.215926	6.142×10^{-4}
2.8	13.553885	13.554286	4.006×10^{-4}
2.9	13.927046	13.927241	1.953×10^{-4}
3.0	14.333333	14.333333	

Finite-difference method for nonlinear problems

The error order again can be improved by Richardson's extrapolation: solve the problem with $h = 0.1$, 0.05, and 0.025, and then use extrapolation as before. Accuracy can be improved from 10^{-3} to 10^{-10}.

Rayleigh-Ritz method

Idea: Convert the BVP to an integral minimization problem, and then find the minimizer from the function space spanned by a set of basis functions.

We consider a standard BVP with second-order ODE:

$$\begin{cases} -\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right) + q(x)y = f, \quad 0 \le x \le 1 \\ y(0) = 0, \ y(1) = 0 \end{cases}$$

Problems with general interval $[a, b]$ and boundary conditions $y(a) = \alpha, y(b) = \beta$ can be converted into the standard one above.

For example, if $y(0) = \alpha, y(1) = \beta$, then set $z(x) = y(x) - ((1 - x)\alpha + x\beta)$ and derive the ODE of z with boundary value $z(0) = z(1) = 0$.

Rayleigh-Ritz method

Theorem (Variational form of BVP)

Suppose $p \in C^1, q, f \in C$, $p \geq \delta$ *for some* $\delta > 0$ *and* $q \geq 0$ *on* $[0, 1]$, *and* $y \in C_0^2$, *then* y *is the unique solution to*

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right) + q(x)y = f, \quad 0 \leq x \leq 1 \quad \underline{\mathit{ODE}}$$

if and only if y *is the unique function that minimizes* $I[\cdot]$ *where*

$$I[u] = \int_0^1 \left(p(x)[u'(x)]^2 + q(x)[u(x)]^2 - 2f(x)u(x)\right) \mathrm{d}x \quad \underline{\mathit{Energy}}$$

Rayleigh-Ritz method

Proof.

1. A solution y to (ODE) satisfies:

$$\int_0^1 f(x)u(x)dx = \int_0^1 p(x)\frac{dy}{dx}(x)\frac{du}{dx}(x)+q(x)y(x)u(x)dx, \quad \forall u \in C_0^1[0,1] \quad \underline{\mathit{Weak}}$$

   This can be verified by multiplying u on both sides of ODE, taking integral, and integrating by part.

2. y minimizes Energy iff y satisfies Weak: For any $y, u \in C_0^1[0,1]$, define $g(\epsilon) = I[y + \epsilon u]$, then $g''(\epsilon) \geq 0$, so I is a convex functional. Therefore y minimizes Energy iff $g'(0) = 0$ for all u (i.e., y satisfies Weak).

3. Weak admits at most one solution: if y_1, y_2 both satisfies Weak, then $y = y_1 - y_2$ satisfies Weak with $f = 0$, i.e., y minimizes $J[u] = \int_0^1 (p(u')^2 + qu^2)dx$. Hence $y \equiv 0$ (since $J[u] \geq 0$ and $= 0$ only if $u \equiv 0$).

□

Rayleigh-Ritz method

Now we know BVP is equivalent to an energy minimization problem:

$$I[u] = \int_0^1 \Big(p(x)[u'(x)]^2 + q(x)[u(x)]^2 - 2f(x)u(x) \Big)\,\mathrm{d}x$$

Steps of Rayleigh-Ritz method:

1. Create a set of basis functions $\{\phi_i \mid 1 \le i \le n\}$, and set approximation $\phi = \sum_i c_i\phi_i$ to $y = \mathrm{argmin}_u\, I[u]$.
2. Find c by minimizing $I[\phi] = I[\sum_i c_i\phi_i]$, i.e., $\partial_{c_i} I[\sum_i c_i\phi_i] = 0$ for all i.

Rayleigh-Ritz method

Step 2 above yields a linear **normal equation** of c, denoted by $Ac = b$, where $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ with

$$a_{ij} = \int_0^1 \left[p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x) \right] dx$$

$$b_i = \int_0^1 f(x)\phi_i(x)dx$$

Once c is solved, the minimizer of I can be set to $\phi = \sum_i c_i\phi_i$.

Now the key is the design of basis functions in Step 1. If properly designed, A will be a band matrix (and even tridiagonal matrix).

Piecewise-linear basis

Steps to create a piecewise linear basis:

1. Partition $[0, 1]$ into $n + 1$ subintervals:

$$0 = x_0 < x_1 < \cdots < x_{n+1} = 1$$

   Step size $h_i = x_{i+1} - x_i$ for $i = 0, \dots, n$.

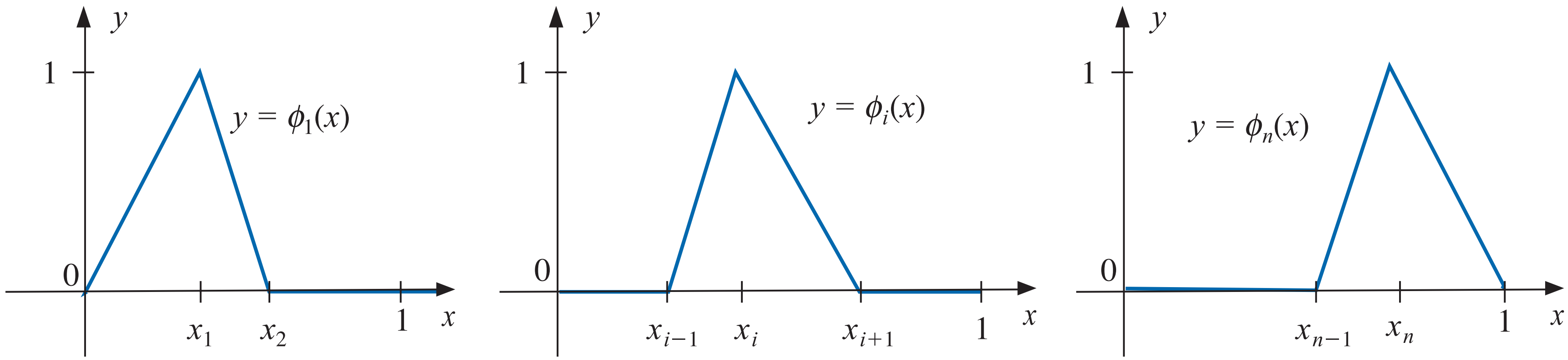
2. Set $\{\phi_i\}$ for $i = 1, \dots, n$ by:

$$\phi_i(x) = \begin{cases} 0, & \text{if } 0 \le x \le x_{i-1} \\ \frac{1}{h_{i-1}} \left(x - x_{i-1}\right), & \text{if } x_{i-1} < x \le x_i \\ \frac{1}{h_i} \left(x_{i+1} - x\right), & \text{if } x_i < x \le x_{i+1} \\ 0, & \text{if } x_{i+1} < x \le 1 \end{cases}$$

Piecewise linear basis

Namely, $\phi_i(x)$ is 1 at $x = x_i$ and linearly decays to 0 at $x = x_{i\pm1}$, then stays as 0 outside of $[x_{i-1}, x_{i+1}]$.

Example of piecewise linear basis functions:



Piecewise linear basis

Several properties about piecewise linear basis:

1. ϕ_i is differentiable except at x_{i-1}, x_i, x_{i+1}:

$$\phi_i'(x) = \begin{cases} 0, & \text{if} \quad 0 < x < x_{i-1} \\ \frac{1}{h_{i-1}}, & \text{if} \quad x_{i-1} < x < x_i \\ -\frac{1}{h_i}, & \text{if} \quad x_i < x < x_{i+1} \\ 0, & \text{if} \quad x_{i+1} < x < 1 \end{cases}$$

2. ϕ_i and ϕ_j do not interfere if $|i - j| > 1$:

$$\phi_i(x)\phi_j(x) \equiv 0 \quad \text{and} \quad \phi_i'(x)\phi_j'(x) \equiv 0$$

   Hence $A = [a_{ij}]$ in the normal equation $Ac = b$ is a tridiagonal matrix.

Piecewise linear basis

$$
\begin{aligned}
a_{ii} &= \int_0^1 \left\{ p(x) \left[\phi_i'(x)\right]^2 + q(x) \left[\phi_i(x)\right]^2 \right\} dx \\
&= \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x) dx + \left(\frac{-1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} p(x) dx \\
&\quad + \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} \left(x - x_{i-1}\right)^2 q(x) dx + \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} \left(x_{i+1} - x\right)^2 q(x) dx \\
a_{i,i+1} &= \int_0^1 \left\{ p(x)\phi_i'(x)\phi_{i+1}'(x) + q(x)\phi_i(x)\phi_{i+1}(x) \right\} dx \\
&= -\left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} p(x) dx + \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} \left(x_{i+1} - x\right)\left(x - x_i\right) q(x) dx \\
a_{i,i-1} &= \int_0^1 \left\{ p(x)\phi_i'(x)\phi_{i-1}'(x) + q(x)\phi_i(x)\phi_{i-1}(x) \right\} dx \\
&= -\left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x) dx + \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} \left(x_i - x\right)\left(x - x_{i-1}\right) q(x) dx \\
b_i &= \int_0^1 f(x)\phi_i(x) dx = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} \left(x - x_{i-1}\right) f(x) dx + \frac{1}{h_i} \int_{x_i}^{x_{i+1}} \left(x_{i+1} - x\right) f(x) dx
\end{aligned}
$$

Piecewise linear bassis

There are $6n$ integrals to evaluate:

$$Q_{1,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)\, q(x) dx, \quad \text{for each } i = 1, 2, \ldots, n-1$$

$$Q_{2,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\, q(x) dx, \quad \text{for each } i = 1, 2, \ldots, n$$

$$Q_{3,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2\, q(x) dx, \quad \text{for each } i = 1, 2, \ldots, n$$

$$Q_{4,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x) dx, \quad \text{for each } i = 1, 2, \ldots, n+1$$

$$Q_{5,i} = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} (x - x_{i-1})\, f(x) dx, \quad \text{for each } i = 1, 2, \ldots, n$$

$$Q_{6,i} = \frac{1}{h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - x)\, f(x) dx, \quad \text{for each } i = 1, 2, \ldots, n$$

Piecewise linear basis

Then A and b are computed as

$$\begin{aligned}
a_{i,i} &= Q_{4,i} + Q_{4,i+1} + Q_{2,i} + Q_{3,i}, \qquad \text{for each } i = 1, 2, \dots, n \\
a_{i,i+1} &= -Q_{4,i+1} + Q_{1,i}, \qquad \text{for each } i = 1, 2, \dots, n-1 \\
a_{i,i-1} &= -Q_{4,i} + Q_{1,i-1}, \qquad \text{for each } i = 2, 3, \dots, n \\
b_i &= Q_{5,i} + Q_{6,i}, \qquad \text{for each } i = 1, 2, \dots, n
\end{aligned}$$

We can show that A is positive definite.

Piecewise linear basis

Two ways to approximate the $6n$ integrals Q's:

1. Quadratures such as Simpson's rule.
2. Approximate p, q, r by piecewise linear functions and compute integrals. For example, $p(x) \approx \sum_i p(x_i)\phi_i(x)$ etc., then

$$
\begin{aligned}
Q_{1,i} &\approx \frac{h_i}{12}[q(x_i) + q(x_{i+1})] \\
Q_{2,i} &\approx \frac{h_{i-1}}{12}\left[3q\left(x_i\right) + q\left(x_{i-1}\right)\right], \\
Q_{3,i} &\approx \frac{h_i}{12}\left[3q\left(x_i\right) + q\left(x_{i+1}\right)\right] \\
Q_{4,i} &\approx \frac{h_{i-1}}{2}\left[p\left(x_i\right) + p\left(x_{i-1}\right)\right] \\
Q_{5,i} &\approx \frac{h_{i-1}}{6}\left[2f\left(x_i\right) + f\left(x_{i-1}\right)\right] \\
Q_{6,i} &\approx \frac{h_i}{6}\left[2f\left(x_i\right) + f\left(x_{i+1}\right)\right]
\end{aligned}
$$

Each approximation has error order $O(h_i^3)$.

Piecewise linear basis

Example (Rayleigh-Ritz method with piecewise linear basis)

Solve the BVP below using Rayleigh-Ritz method and piecewise linear basis with $h_i = h = 0.1$:

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \quad 0 \le x \le 1, \quad y(0) = 0,\ y(1) = 0$$

Solution: We have $p(x) \equiv 1$, $q(x) \equiv \pi^2$, $f(x) = 2\pi^2 \sin(\pi x)$. Then apply the formula above to obtain $Q_{1,i}, \dots, Q_{6,i}$ for $i = 0, \dots, 9$, and then A and b. Then solve c from $Ac = b$, and obtain $\phi(x) = \sum_i c_i \phi_i(x)$ (note that $\phi(x)$ is piecewise linear function and $\phi(x_i) = c_i$).

Piecewise linear basis

i	x_i	$\phi(x_i)$	$y(x_i)$	$\lvert\phi(x_i) - y(x_i)\rvert$
1	0.1	0.3102866742	0.3090169943	0.00127
2	0.2	0.5902003271	0.5877852522	0.00241
3	0.3	0.8123410598	0.8090169943	0.00332
4	0.4	0.9549641896	0.9510565162	0.00390
5	0.5	1.0041087710	1.0000000000	0.00411
6	0.6	0.9549641893	0.9510565162	0.00390
7	0.7	0.8123410598	0.8090169943	0.00332
8	0.8	0.5902003271	0.5877852522	0.00241
9	0.9	0.3102866742	0.3090169943	0.00127

The error order is $O(h^2)$ due to the nature of linear (first-order) approximation of the integand.

B-spline basis

We can create C^2 basis functions using the idea of cubic splines. These are called the **B-splines** (basis splines).

We start from the cubic spline function S:

$$S(x) = \begin{cases} 0, & \text{if } x \le -2 \\ \frac{1}{4}(2+x)^3, & \text{if } \ -2 \le x \le -1 \\ \frac{1}{4}\left[(2+x)^3 - 4(1+x)^3\right], & \text{if } \ -1 < x \le 0 \\ \frac{1}{4}\left[(2-x)^3 - 4(1-x)^3\right], & \text{if } \ 0 < x \le 1 \\ \frac{1}{4}(2-x)^3, & \text{if } \ 1 < x \le 2 \\ 0, & \text{if } \ 2 < x \end{cases}$$

B-spline basis

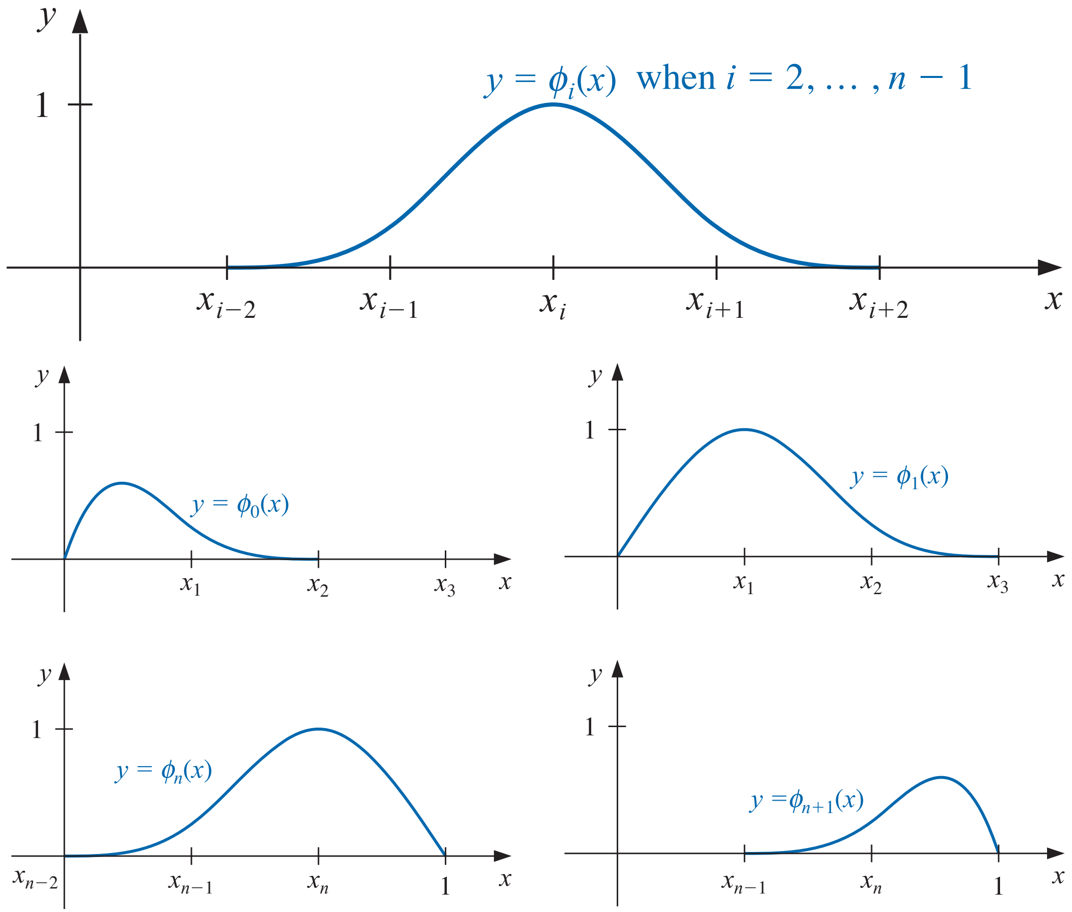
Then construct B-spline basis functions $\{\phi_i \mid 0 \le i \le n+1\}$:

$$
\phi_i(x) = \begin{cases}
S\left(\frac{x}{h}\right) - 4S\left(\frac{x+h}{h}\right), & \text{if } i = 0 \\
S\left(\frac{x-h}{h}\right) - S\left(\frac{x+h}{h}\right), & \text{if } i = 1 \\
S\left(\frac{x-ih}{h}\right), & \text{if } 2 \le i \le n-1 \\
S\left(\frac{x-nh}{h}\right) - S\left(\frac{x-(n+2)h}{h}\right), & \text{if } i = n \\
S\left(\frac{x-(n+1)h}{h}\right) - 4S\left(\frac{x-(n+2)h}{h}\right), & \text{if } i = n+1
\end{cases}
$$

- $\phi_i \in C_0^2[0,1]$.
- $\{\phi_i\}$ are independent.

B-spline basis

$\phi_i(x)$ for $2 \leq i \leq n-1$ (top) and $\phi_0, \phi_1, \phi_n, \phi_{n+1}$ (bottom four).



B-spline basis

Let $\phi(x) = \sum_i c_i \phi_i(x)$. Then the normal equation $\partial_c I[\phi] = 0$ is $Ac = b$ where $A = [a_{ij}]$ is a positive definite band matrix with bandwidth ≤ 7, where

$$a_{ij} = \int_0^1 \left\{ p(x)\phi_i'(x)\phi_j'(x) + q(x)\phi_i(x)\phi_j(x) \right\} dx$$

$$b_i = \int_0^1 f(x)\phi_i(x) dx$$

To compute these integrals, we can replace p, q, f by their cubic spline interpolations (so on each subinterval they are cubic polynomials), and integrals can be evaluated exactly (as the integrands are polynomials).

B-spline basis

Example (Rayleigh-Ritz with B-spline basis)

Solve the BVP below using Rayleigh-Ritz method and B-spline basis with $h_i = h = 0.1$:

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \quad 0 \le x \le 1, \quad y(0) = 0, \; y(1) = 0$$

Solution: We have $p(x) \equiv 1$, $q(x) \equiv \pi^2$, $f(x) = 2\pi^2 \sin(\pi x)$. Then approximate $Q_{1,i}, \dots, Q_{6,i}$ for $i = 0, \dots, 9$, and then A and b. Then solve c from $Ac = b$, and obtain $\phi(x) = \sum_i c_i \phi_i(x)$.

B-spline basis

Numerical result:

i	c_i	x_i	$\phi(x_i)$	$y(x_i)$	$\lvert y(x_i) - \phi(x_i) \rvert$
0	$0.50964361 \times 10^{-5}$	0	0.00000000	0.00000000	0.00000000
1	0.20942608	0.1	0.30901644	0.30901699	0.00000055
2	0.39835678	0.2	0.58778549	0.58778525	0.00000024
3	0.54828946	0.3	0.80901687	0.80901699	0.00000012
4	0.64455358	0.4	0.95105667	0.95105652	0.00000015
5	0.67772340	0.5	1.00000002	1.00000000	0.00000020
6	0.64455370	0.6	0.95105130	0.95005520	0.00000061
7	0.54828951	0.7	0.80901773	0.80901699	0.00000074
8	0.39835730	0.8	0.58778690	0.58778525	0.00000165
9	0.20942593	0.9	0.30901810	0.30901699	0.00000111
10	$0.74931285 \times 10^{-5}$	1.0	0.00000000	0.00000000	0.00000000

This is much more accurate than the one with piecewise linear basis.