HW 10

ankit gupta

July 2023

1 Linear Shooting Method for a Two-point Boundary Value Problem

1.1 a

Given the differential equation:

$$y''(x) = y'(x) + 2y(x) + \cos(x), \quad 0 \leq x \leq \frac{\pi}{2}$$

and the boundary conditions:

$$y(0) = -0.3, \quad y\left(\frac{\pi}{2}\right) = -0.1$$

The proposed solution is:

$$y(x) = -\frac{\sin(x) + 3\cos(x)}{10}$$

To confirm if this is a solution, we compute the first and second derivatives of $y(x)$:

$$y'(x) = -\frac{1}{10}(\cos(x) - 3\sin(x))$$

$$y''(x) = -\frac{1}{10}(-\sin(x) - 3\cos(x))$$

Now, substitute $y(x)$, $y'(x)$, and $y''(x)$ into the differential equation:

$$y''(x) - y'(x) - 2y(x) = \cos(x)$$

$$-\frac{1}{10}(-\sin(x) - 3\cos(x)) - (-\frac{1}{10}(\cos(x) - 3\sin(x))) - 2 \cdot -\frac{\sin(x) + 3\cos(x)}{10} = \cos(x)$$

This simplifies to:

$$\sin(x) + 3\cos(x) + \cos(x) - 3\sin(x) - 2\sin(x) - 6\cos(x) = \cos(x)$$

Thus, the given function $y(x)$ is a solution of the differential equation. Checking the boundary conditions:

$$y(0) = -\frac{\sin(0) + 3\cos(0)}{10} = -0.3$$

$$y\left(\frac{\pi}{2}\right) = -\frac{\sin\left(\frac{\pi}{2}\right) + 3\cos\left(\frac{\pi}{2}\right)}{10} = -0.1$$

The proposed solution satisfies both the differential equation and the boundary conditions, therefore, we can conclude that $y(x) = -\frac{\sin(x)+3\cos(x)}{10}$ is indeed the exact solution of the given two-point boundary value problem.

1.2 b

check matlab file hw10.m

2 More Practice on Linear Shooting Method

2.1 a

The given differential equation is $u''(x)+6u(x) = 6x^3$, with boundary conditions $u(0) = 0$, and $u(1) + u'(1) = 2$.

The proposed solution is $u(x) = x^3 - x$. To verify this, compute the first and second derivatives of $u(x)$:

$u'(x) = 3x^2 - 1$

$u''(x) = 6x$

Substitute $u(x)$, $u'(x)$, and $u''(x)$ into the differential equation:

$u''(x) + 6u(x) = 6x^3$

$6x + 6(x^3 - x) = 6x^3$

Simplifying the left hand side gives $6x^3$, which is equal to the right hand side. Thus, $u(x) = x^3 - x$ satisfies the differential equation.

Next, check the boundary conditions:

$u(0) = 0^3 - 0 = 0$

$u'(1) = 3 * 1^2 - 1 = 2$

$u(1) + u'(1) = (1^3 - 1) + 2 = 2$

So, $u(x) = x^3 - x$ satisfies the boundary conditions, and is therefore the exact solution of the given two-point boundary value problem.

2.2 b

Consider the two-point boundary value problem for the unknown $u(x)$ given by

$$u'' + 6u = 6x^3, \quad u(0) = 0, \quad u(1) + u'(1) = 2.$$

To solve this problem using the shooting method, we proceed as follows:

1. First, we rewrite the second-order boundary value problem as a system of two first-order differential equations. We introduce a new variable $v(x)$ and let $v = u'$. Our system is now v' = 6x$^3 - 6u$, $\quad with \quad v(0) = s$, $u' = v$, $\quad with \quad u(0) = 0$, where s is an initial guess for the derivative at the initial point.

2. We choose an initial guess for s. This choice may depend on the problem and can sometimes be guided by physical intuition or prior knowledge. In the absence of such guidance, it is common to start with a guess of $s = 0$.

3. We solve the initial value problem for $u(x)$ and $v(x)$ from $x = 0$ to $x = 1$ using a method like the fourth-order Runge-Kutta method.

4. We check whether $u(1) + v(1) = 2$. If it is, then we have found the correct solution. If it isn't, we need to adjust our guess for s.

5. We adjust s using a root-finding technique, such as the secant method or bisection method. The goal is to find a value of s such that $u(1)+v(1) = 2$.

6. We repeat the process from step 3, adjusting s each time, until $u(1) + v(1)$ is sufficiently close to 2 (within a predefined tolerance).

The resulting solution $u(x)$ is an approximation to the true solution of the boundary value problem, and the accuracy depends on the step size used in the initial value problem solver and the tolerance used for the root-finding method.

2.3 c

check matlab file hw10$_2.m$

3 Non-linear Shooting for a Two-point Boundary Value Problem

3.1 a

The given differential equation is $y''(x) = -(y'(x))^2 - y(x) + \cos^2(x)$, with boundary conditions $y(0) = 0$, and $y(\pi) = 0$.

The proposed solution is $y(x) = \sin(x)$. To verify this, compute the first and second derivatives of $y(x)$:

$y'(x) = \cos(x)$

$y''(x) = -\sin(x)$

Substitute $y(x)$, $y'(x)$, and $y''(x)$ into the differential equation:

$y''(x) = -(y'(x))^2 - y(x) + \cos^2(x)$

$-\sin(x) = -\cos^2(x) - \sin(x) + \cos^2(x)$

Simplifying gives $-\sin(x) = -\sin(x)$. Thus, $y(x) = \sin(x)$ satisfies the differential equation.

Next, check the boundary conditions:

$y(0) = \sin(0) = 0$
$y(\pi) = \sin(\pi) = 0$

So, $y(x) = \sin(x)$ satisfies the boundary conditions, and is therefore the exact solution of the given two-point boundary value problem.

3.2 b

check matlab file hw10$_3$.m

4 Finite Difference Method in 1D

4.1 a

We consider the differential equation

$$y'' = y' + 2y + \cos(x), \quad for 0 \leq x \leq \frac{\pi}{2},$$

with boundary conditions $y(0) = -0.3$ and $y\left(\frac{\pi}{2}\right) = -0.1$.

Finite Difference Method

We use the finite difference method to obtain an approximation of the solution. Consider a uniform grid with mesh size $h = \frac{\pi}{2N}$. The approximations for the first and second derivatives at a point x_i are given by

$$y_i' \approx \frac{y_{i+1} - y_{i-1}}{2h}, \quad y_i'' \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Substituting these approximations into the differential equation, we obtain the tridiagonal system of linear equations:

$$-y_{i-1}\left(\frac{2}{h^2} - \frac{1}{2h}\right) + y_i\left(\frac{4}{h^2} - 2\right) - y_{i+1}\left(\frac{2}{h^2} + \frac{1}{2h}\right) = -\cos(x_i),$$

for $i = 1, 2, \ldots, N-1$, along with the boundary conditions $y_0 = -0.3$ and $y_N = -0.1$.

This system can be written in matrix form as $\mathbf{AY} = \mathbf{B}$, where $\mathbf{A}$ is a $(N+1) \times (N+1)$ tridiagonal matrix, $\mathbf{Y}$ is the vector of unknowns y_i, and $\mathbf{B}$ is a vector whose ith entry is $-\cos(x_i)$ for $i = 1$ to $N-1$, and whose first and last entries are given by the boundary conditions -0.3 and -0.1, respectively.

4.2 b

check matlab file hw10$_4$.m