Boundary value problems

Andreas Hegseth

April 17, 2022

1 problem

solve the second-order boundary value problem: y = y + 2y + cos(x) , $0 \leq x \leq \pi$ 2 , y(0) = 0.1 , $y(\pi/2)$ = -0.3 . (1) This problem has exact solution y(x) = 1 10 (sin(x)+3 cos(x)). Your task is to approximate the solution of this linear boundary value problem using the Shooting Method and the Finite Difference Method. For each method, try $h = \pi/16$ 8 and $h = \pi/8$ 16 and characterize how your approximation compares to the exact solution.

2 mathematics

Boundary value problems Theorem 1) Suppose the function f in the BVP y" = f(x,y,y'), for a $\leq x \leq b, with$y(a)=α and $y(b) = \beta$, is continuous on the set $D = [(x, y, y')|for a \leq x \leq b, with -\infty < y < \infty and -\infty < y', \infty]$, and the partial derivative f_y and f_y' are also continuous on D. If

i)$f_y(x, y, y') > 0, forall(x, y, y')\epsilon D$, and

ii) a constant M exist, with $|f_{y')|\leq M}$ for all (x,y,y') D, then the BVP has a unique solution. Our first method is called the shooting method we treat in like an ivp with the boundary condition of $y(a) = \alpha$ and $y'(a) = t$ where t must be chosen so that the solution satisfies the remaining boundary condition, y(b) = . Since t, being the first derivative of y(x) at x = a, is the "initial slope" of the solution, this approach requires selecting the proper slope, or "trajectory", so that the solution will "hit the target" of y(x) = at x = b. This viewpoint indicates how the shooting method earned its name. Note that since the ODE associated with the IVP is of second-order, it must normally be rewritten as a system of first-order equations before it can be solved by standard numerical methods such a forward, backward and central Euler.

Finite Difference Method This method is a second-order BVP, where y"=p(x)y'+q(x)y+r(x), for $a \leq x \leq b$, with $y(a) = \alpha$ and $y(b) =$, requires that difference-quotient approximations be used to approximate both y' and y". First, we select an integer $N > 0$ and divide the interval [a,b] into (N+1) equal subintervals whose endpoints are the mesh points $x_i = a+ih$, for $i = 0, 1..., N+1$, where $h = (b-a)/(N+1)$. Choosing the side step h in this manner facilities the applications of a matrix, which solves a linear system involving an NXN matrix. At the interior mesh points x_i, for i=1,2,...,N, the differential equation to be approximated is $y"(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i)$ (1)* We expand y into a third Taylor polynomial about x_i evaluated at x_{i+1} and x_{i-1}, we have, assuming that $yC^4[x_{i-1}, x_{i+1}]$, $y(x_{i+1}) = y(x_i + h) = y(x_i) + h * y'(x_i) + [h^2/2]y"(x_i) + [h^3/6]y'''(x_i) + [h^4/24]y^4(\xi_i^+)$ for some $\xi_i^+ in(x_i, x_{i+1})$ and $y(x_{i-1}) = y(x_i-h) = y(x_i)-h*y'(x_i)+[h^2/2]y"(x_i)-[h^3/6]y'''(x_i)+[h^4/24]y^4(\xi_i^-)$ for some $\xi_i^- in(x_{i-1})$. If these equations are added, we have $y(x_{i+1}) + y(x_{i-1}) = 2y(x_i) + h^2 * y"(x_i) + [h^4/24](y^4(\xi_i^+) + y^4(\xi_i^-))$ ans solving for $y"(x_i) = (1/h^2)[y(_{i+1}) - 2y(x_i) + y(x_{i-1})] - (h^2/24)[y^4(\xi_i^+) + y^4(\xi_i^-)]$. from our theorem we can simplify the error term to give $y"(x_i) = (1/h^2)[y(x_{i+1}) - 2y(x_i) + y(x_i-1)] - (h^2/12)y^4(\xi_i)$ for some $\xi_i in(x_{i-1}, x_{i+1})$. This is called the centered-difference formula for $y"(x_i)$. from centered-difference formula we can get $y'(x_i) = (1/2h)[y(x_{i+1}) - y(x_{i-1})] - (h^2/6)y'''(\eta_i)$, for some $\eta_i in(x_{i-1}, x_{i+1})$. The use of these centered-difference in formula 1 we get $[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})]/h^2 = p(x_i)[(y(x_{i+1}) - y(x_{i-1})/2h] + q(x_i)y(x_i) + r(x_i) - (h^2/12)[2p(x_i)y'''(\eta) - y^4(\xi_i)]$. A Finite-Difference method with truncation error of order $O(h^2)$ results by using this equation together with the boundary $y(a) = \alpha$ and $y(b) =$ to define the system of linear equations $w_0 = \alpha$, $w_{N+1} =$ and (2)* $((-w_{i+1} + 2w_i - w_{i-1})/h^2) + p(x_i)((w_{i+1} - w_{i-1})/2h) + q(x_i)w_i = -r(x_i)$ for i=1,2,...,N. From

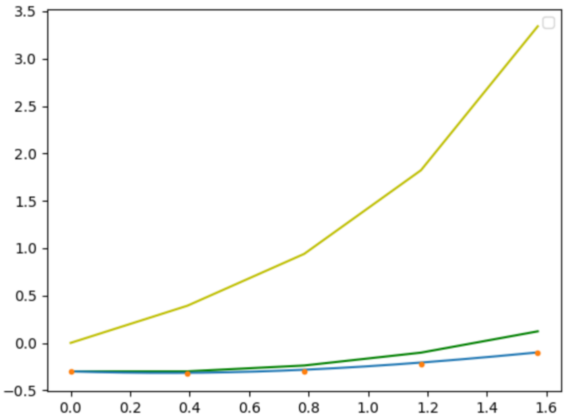


Figure 1: shooting method using h= $\pi/8$

(2)* we can rewrite it as $-(1+(h/2)p(x_i))w_{i-1}+(2+h^2q(x_i))w_i-(1-(h/2)p(x_i))w_{i+1}=-h^2r(x_i)$, then we get the resulting matrix that is tridiagonal NXN in Ax=b where

$$\text{A}=\begin{bmatrix} 2+h^2q(x_1) & -1+(h/2)p(x_1) & o & . & . & ..0 \\ -1+(h/2)p(x_1) & 2+h^2q(x_1) & -1+(h/2)p(x_1) & . & . & ..0 \\ 0 & .. & .. & . & . & 0 \\ 0 & .. & .. & .. & .. & -1+(h/2)p(x_1) \\ 0 & 0 & 0 & 0 & -1+(h/2)p(x_1) & 2+h^2q(x_1) \end{bmatrix}$$

$$\text{W}=\begin{bmatrix} W1 \\ W2 \\ . \\ . \\ W_{N-1} \\ W_N \end{bmatrix}, b=\begin{bmatrix} -h^2r(x_1)+(1+(h/2)p(x_1))w_0 \\ -h^2r(x_2) \\ . \\ . \\ . \\ -h^2r(x_{N-1}) \\ -h^2r(x_N)+(1-(h/2)p(x_N))w_{N+1} \end{bmatrix}$$

3 algorithm

shooting method

step 1 import numpy and matplotlib

step 2 define the exact solution and two ivp's

step 3 using an ivp solver using forward Euler

step 4 plot all functions

Finite-difference method

step 1 import numpy and matplotlib and set bounds

step 2 set up matrix A and vector b

step 3 solve using linalg.solver and plot the exact and approximation

4 results

shooting method look at figures 1 and 2

finite-difference look at figures 3 and 4

5 conclusion

With my shooting method code my two ivp's I made are shown as yellow and green. The point are the approximations on the exact. With my finite difference code I don't know why I can't seem to get

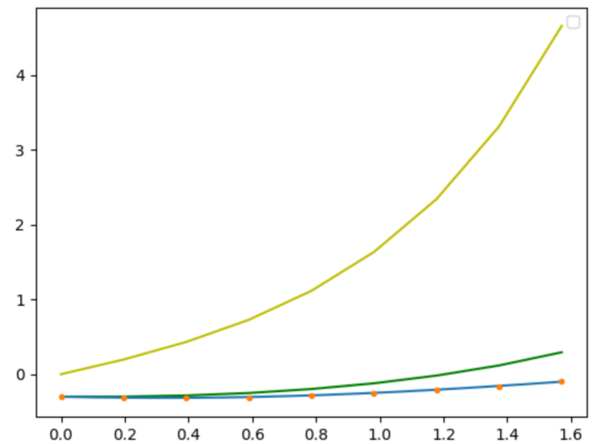
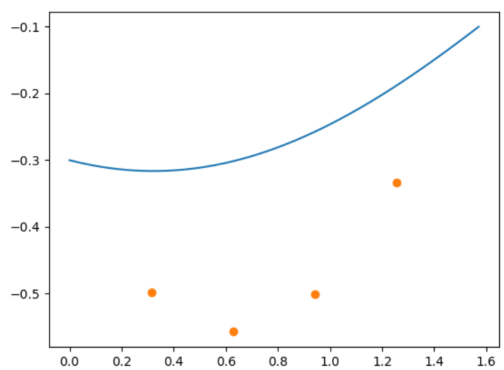
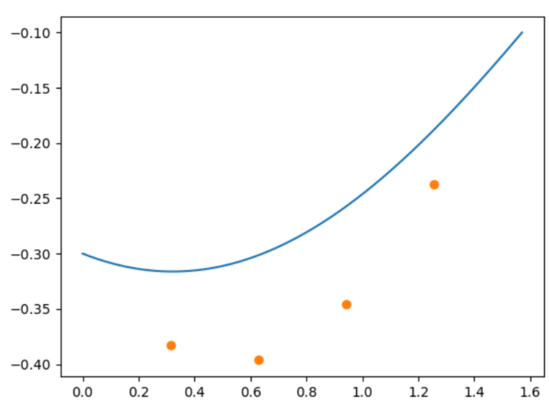


Figure 2: shooting method using h= $\pi/16$



error:0.4231791309100695

Figure 3: finite difference method using h= $\pi/8$



error:0.15422694669240009

Figure 4: finite difference method using h= $\pi/16$

it more accurate. I believe that I made a algebraic error or coding error but I am not sure. Beside that these method for solving BVP are very creative in how to solve them with shooting method the smaller the h it seem to have a greater distance in the ivps though I see a similarity with accuracy's. With finite difference though I don't know because I can seem to get my code to work properly. I do know however that our n depends on our h to be able to fit it in our bounds. I believe the smaller our h the more accurate it would be. All that being said I do believe that these method are very valuable though finite is extremely frustrating to me I do believe it is invaluable.