5.8

Applied Exercises

5

5. Suppose a wolf is chasing a rabbit. The path of the wolf toward the rabbit is called a curve of pursuit. Assume the wolf runs at the constant speed α and the rabbit at the constant speed β. Let the wolf begin at time $t = 0$ at the origin and the rabbit at the point $(0, 1)$. Assume the rabbit runs up the line $x = 1$. Let $(x(t), y(t))$ denote the position of the wolf at time t.

The differential equation describing the curve of pursuit is

$$\frac{dy}{dx} = \frac{1}{2}\left[(1-x)^{-\beta/\alpha} - (1-x)^{\beta/\alpha}\right]$$

5.11

Theoretical Exercises

11

11. The Backward Euler one-step method is defined by

$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}), \quad \text{for } i = 0, \ldots, N-1.$$

Show that $Q(h\lambda) = 1/(1 - h\lambda)$ for the Backward Euler method.

$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}), \text{ for } i = 0, \ldots, N-1$$

Apply it to $\frac{dy}{dt} = \lambda \cdot y$:

$$w_{i+1} = \left(\frac{1}{1 - h\lambda}\right) w_i$$

Substitute $f(t_i, w_i) = \lambda \cdot w_i$

$$w_{i+1} = w_i + h \cdot f(t_{i+1}, w_{i+1})$$
$$w_{i+1} = w_i + h\lambda w_{i+1}$$
$$w_{i+1} - h\lambda w_{i+1} = w_i$$
$$w_{i+1}(1 - h\lambda) = w_i$$

$$w_{i+1} = \left(\frac{1}{1 - h\lambda}\right) w_i$$
$$\Rightarrow w_{i+1} = Q(h\lambda) w_i$$
$$\Rightarrow Q(h\lambda) = \left(\frac{1}{1 - h\lambda}\right)$$

11.1

5

5. Use the Linear Shooting Algorithm to approximate the solution $y = e^{-10x}$ to the boundary-value problem

$$y'' = 100y, \quad 0 \le x \le 1, \quad y(0) = 1, \quad y(1) = e^{-10}.$$

Use $h = 0.1$ and 0.05.

Write the linear boundary value Problem $y'' = 100y, 0 \le x \le 1$ with boundary conditions $y(0) = 1$ and $y(1) = e^{-10}$
The actual solution is $y(x) = e^{-10x}$.
Choose h=0.1 so N=10.
The Linear Shooting Method:
If $y'' = p(x) \cdot y' + q(x) \cdot y + r(x)$ for $a \le x \le b$ with $y(a) = \alpha$ and $y(b) = \beta$ is a Linear boundary value problem and if $y_1(x)$ represents the solution of $y'' = p(x) \cdot y' + q(x) \cdot y + r(x)$ with $a \le x \le b, y(a) = \alpha, y(a) = 0$ and $y_2(x)$ represents the solution of $y'' = p(x) \cdot y' + q(x) \cdot y_{\text{with}}$ $a \le x \le b, y(a) = 0, y(a) = 1$ respectively, then the solution is

$$y(x) = y_1(x) + \left[\frac{\beta - y_1(b)}{y_2(b)}\right] y_2(x) \text{ where } y_2(b) \neq 0$$

I	X(I)	W(1,I)	W(2_I)	Y_actual
0	0.00000000	1.00000000	-10.00000002	1
1	0.10000000	0.37500000	-3.75000002	0.3678794412
2	0.20000000	0.14062499	-1.40625006	0.1353352832
3	0.30000000	0.05273436	-0.52734392	0.04978706837
4	0.40000000	0.01977534	-0.19775435	0.01831563889
5	0.50000000	0.00741563	-0.07415888	0.006737946999
6	0.60000000	0.00278054	-0.02781256	0.002478752177
7	0.70000000	0.00104209	-0.01043915	0.00091 18819656
8	0.80000000	0.00038819	-0.00393811	0.0003354626279
9	0.90000000	0.00013976	-0.00154237	0.0001234098041
10	1.00000000	0.00004540	-0.00074600	0.00004539992976

Now chose h=0.05 so N=20

Numerical Analysis 2 – Amirhossein Mahmoudi – HW3

LINEAR SHOOTING METHOD OUTPUT

I	X(l)	W(I,1)	W(I,2)	Y_actual
0	0.00000000	1.00000000	-10.00000001	1
1	0.05000000	0.60677083	-6.06770835	0.6065306597
2	0.10000000	0.3681/084	-3.68170848	0.3678794412
3	0.15000000	0.22339533	-2.23395333	0.2231301601
4	0.20000000	0.13554977	.1,35549773	0.1353352832
5	0.25000000	0.08224764	-082247653	0.08208499862
6	0.30000000	0.04990547	049905488	0.04978706837
7	035000000	0.03028118	-0.30281202	0.03019738342
8	0.40000000	0.01837371	-0.18373767	0.01831563889
9	0.45000000	0.01114862	-0.11148684	0.01110899654
10	05000000	0.00676462	-0.06764744	0.006737946999
11	0.55000000	0.00410451	-004104695	0.004086771438
12	0.60000000	0.00249033	002490667	0.002478752177
13	0.65000000	0.00151077	-0.01511427	0.001503439193
14	0.70000000	0.00091636	-0.00917237	0.0009118819656
15	0.75000000	0.00055553	-0.00557070	0.0005530843701
16	0.80000000	0.00033715	-0.00338852	0.0003354626279
17	0.85000000	0.00020414	-000207865	0.0002034683690
18	0.90000000	0.00012271	-000129293	0.0001234098041
19	0.95000000	0.00007154	-0.00082459	0.00007485182989
20	1.00000000	0.00004540	-0.00054600	0.00004539992976

This output shows the approximate solutions.

11.2

4.a

4. Use the nonlinear shooting method with $TOL = 10^{-4}$ to approximate the solution to the following boundary-value problems. The actual solution is given for comparison to your results.

a. $y'' = y^3 - yy'$, $1 \le x \le 2$, $y(1) = \frac{1}{2}$, $y(2) = \frac{1}{3}$; use $h = 0.1$; actual solution $y(x) = (x+1)^{-1}$.

b. $y'' = 2y^3 - 6y - 2x^3$, $1 \le x \le 2$, $y(1) = 2$, $y(2) = \frac{5}{2}$; use $h = 0.1$; actual solution $y(x) = x + x^{-1}$.

c. $y'' = y' + 2(y - \ln x)^3 - x^{-1}$, $2 \le x \le 3$, $y(2) = \frac{1}{2} + \ln 2$, $y(3) = \frac{1}{3} + \ln 3$; use $h = 0.1$; actual solution $y(x) = x^{-1} + \ln x$.

d. $y'' = 2(y')^2 x^{-3} - 9y^2 x^{-5} + 4x$, $1 \le x \le 2$, $y(1) = 0$, $y(2) = \ln 256$; use $h = 0.05$; actual solution $y(x) = x^3 \ln x$.

(a)

This is of the form $y'' = f(x, y, y')$ for $a \le x \le b$ with $y(a) = a$ and $y'(a) = t$

Choose the parameter $t = t_k$ such that $\lim_{k\to\infty} y(b, t_k) = y(b) = \beta$ where $y(x, t_k)$ denotes the solution to a nonlinear initial value problem involving a parameter t with $t = t_k$ and $y(x)$ denotes the solution to a nonlinear boundary value problem.

$$h = 0.1, N = 10, TOL = 10^{-4}, M = 100$$

It is required to approximate the solution to the following initial value problem involving a parameter t

With $t = t_k$:

$$y'' = y^3 - yy', 1 \le x \le 2 , y(1) = 0 , y'(1) = t_k$$

$$z'' = \left(\frac{\partial f}{\partial y}\right) z + \left(\frac{\partial f}{\partial y'}\right) z' = (3y^2 - y')z - yz', \ \ 1 \le x \le 2, z(1) = 0, z'(1) = 1$$

I	X(I)	WI(I)	W2(l)	Y_actual
0	1.00000000	0.50000000	-0.24999738	0.5000000000
1	1.10000000	0.47619079	-0.22675486	0.4761904762
2	1.20000000	0.45454606	-0.20660913	0.4545454545
3	1.30000000	0.43478349	-0.18903352	0.4347826087
4	1.40000000	0.41666781	-0.17360875	0.4166666667
5	1.50000000	0.40000140	-0.15999765	0.4000000000
6	1.60000000	038461703	-0.14792664	0.3846153846
7	1.70000000	0.37037226	-0.13717184	0.3703703704
8	1.80000000	0.35714500	-0.12754863	0.3571428571
9	1.90000000	0.34482998	-0.11890365	0.3448275862
10	2.0000000	0.33333597	-0.11110866	0.3333333333

Convergence in 3 iterations

$$t = -24999738E - 01$$

6

6. **a.** Change Algorithm 11.2 to incorporate the Secant method instead of Newton's method. Use $t_0 = (\beta - \alpha)/(b - a)$ and $t_1 = t_0 + (\beta - y(b, t_0))/(b - a)$.

b. Repeat Exercise 4(a) and 4(c) using the Secant algorithm derived in part (a) and compare the number of iterations required for the two methods.

A)

Sudo code:

```
def nonlinear_Shooting_secant_method(f,a,b,alpha,beta,tol,M,N):
    h=(b-a)/N
    k=2
    tk1=(beta-alpha)/(b-a)
    w=[[0, 0] for _ in range(N)]
    w[0][0]=alpha
    w[0][1]=tk1
    ks=[[0]*2]*4
    result=[]
    for i in range(1,N):
        x=a+(i-1)*h
        ks[0][0]=h*w[i-1][1]
        ks[0][1]=h*f(x,w[i-1][0],w[i-1][1])
        ks[1][0]=h*(w[i-1][1]+ks[0][1]/2)
        ks[1][1]=h*f(x+h/2,w[i-1][0]+ks[0][0],w[i-1][1]+ks[0][1]/2)
        ks[2][0]=h*(w[i-1][1]+ks[1][1]/2)
        ks[2][1]=h*f(x+h/2,w[i-1][0]+ks[2][0],w[i-1][1]+ks[2][1]/2)
        ks[3][0]=h*(w[i-1][0]+ks[2][1]/2)
        ks[3][1]=h*f(x+h/2,w[i-1][0]+ks[2][0]+ks[2][0],w[i-1][1]+ks[2][1])
        w[i][0]=w[i-1][0]+(ks[0][0]+2*ks[1][0]+2*ks[2][0]+ks[3][0])/6
        w[i][1]=w[i-1][1]+(ks[0][1]+2*ks[1][1]+2*ks[2][1]+ks[3][1])/6

    tk2=tk1+(beta-w[N-1][0])/(b-a)
    while(k<=M):
        w[0][1]=tk2
        hold=w[N-1][0]
        for i in range(1,N):
            x=a+(i-1)*h
            ks[0][0]=h*w[i-1][1]
            ks[0][1]=h*f(x,w[i-1][0],w[i-1][1])
            ks[1][0]=h*(w[i-1][1]+ks[0][1]/2)
            ks[1][1]=h*f(x+h/2,w[i-1][0]+ks[0][0],w[i-1][1]+ks[0][1]/2)
            ks[2][0]=h*(w[i-1][1]+ks[1][1]/2)
            ks[2][1]=h*f(x+h/2,w[i-1][0]+ks[2][0],w[i-1][1]+ks[2][1]/2)
            ks[3][0]=h*(w[i-1][0]+ks[2][1]/2)
            ks[3][1]=h*f(x+h/2,w[i-1][0]+ks[2][0]+ks[2][0],w[i-1][1]+ks[2][1])
            w[i][0]=w[i-1][0]+(ks[0][0]+2*ks[1][0]+2*ks[2][0]+ks[3][0])/6
            w[i][1]=w[i-1][1]+(ks[0][1]+2*ks[1][1]+2*ks[2][1]+ks[3][1])/6
        if abs(w[N-1][0]-beta)<=tol:
            for i in range(N):
                x=a+i*h
                result.append([x,w[i][0],w[i][1]])
            return result
        tk=tk2-((w[N-1][0]*(tk2-tk1))/(w[N-1][0]-hold))
        tk1=tk2
        tk2=tk
        k=k+1

    print('max interation exceede')
    return result
```

B

NONLINEAR SHOOTING WITH SECANT METHOD

I	X(l)	WI(I)	W2(l)
0	1.00000000	0.50000000	-0.24999738
1	1.10000000	0.47619079	-0.22675486
2	1.20000000	0.45454606	-0.20660913
3	1.30000000	0.43478349	-0.18903352
4	1.40000000	0.41666781	-0.17360875
5	1.50000000	0.40000140	-0.15999765
6	1.60000000	0.38461703	-0.14792664
7	1.70000000	0.37037226	-0.13717184
8	1.80000000	0.35714500	-0.12754863
9	1.90000000	0.34482998	-0.11890365
10	2.00000000	0.33333597	-0.11110866

Convergence in 4 iterations

$$t = -2.4999738e-01$$

11.3

5

5. Use the Linear Finite-Difference Algorithm to approximate the solution $y = e^{-10x}$ to the boundary-value problem

$$y'' = 100y, \quad 0 \le x \le 1, \quad y(0) = 1, \quad y(1) = e^{-10}.$$

Use $h = 0.1$ and 0.05. Can you explain the consequences?

If $y'' = p(x)y' + q(x)y + r(x), a \le x \le b, y(a) = \alpha, y(b) = \beta$ is a linear boundary value problem. To solve this equation, the Linear finite difference method requires the replacement of derivative by the corresponding difference quotient approximations, choosing an integer N > 0 as shown below.

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

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

$x_i = a + ih$ for $i = 0,1,\dots,N+1$

If $y(x_i) = w_i \,, y(x_{i+1}) = w_{i+1} \,, y(x_{i-1}) = w_{i-1}$:

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

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

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

$$for \quad w_0 = \alpha \;, w_{N+1} = \beta \; for\; each\; i = 0,1,2,..N+1$$

The corresponding system of equations is given by $A \cdot w = b$

$$A = \begin{pmatrix} 2 + h^2 q(x_1) & \frac{h}{2}p(x_1) - 1 & \cdots & & 0 & 0 \\ \frac{-h}{2}p(x_2) - 1 & 2 + h^2 q(x_2) & & & & 0 \\ 0 & & \ddots & & & \\ & & & \ddots & & \frac{h}{2}p(x_{N-1}) - 1 \\ 0 & & 0 & & \frac{-h}{2}p(x_N) - 1 & 2 + h^2 q(x_N) \end{pmatrix}$$

Numerical Analysis 2 – Amirhossein Mahmoudi – HW3

$$w = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}, b = \begin{bmatrix} -h^2 r(x_1) + \left(1 + \frac{h}{2} p(x_1)\right) w_0 \\ -h^2 r(x_2) \\ \vdots \\ -h^2 r(x_{N-1}) \\ -h^2 r(x_N) + \left(1 - \frac{h}{2} p(x_N)\right) w_{N+1} \end{bmatrix}$$

The solution to this system yields the approximate solution for the given Linear boundary value problem.

Results for h=0.1:

I	X(I)	W(I)	Y_actual
0	0.00000000	1.00000000	1
1	0.10000000	0.38196601	0.3678794412
2	0.20000000	0.14589802	0.1353352832
3	0.30000000	0.05572807	0.04978706837
4	0.40000000	0.02128617	0.01831563889
5	0.50000000	0.00813045	0.006737946999
6	0.60000000	0.00310518	0.002478752177
7	0.70000000	0.00118509	0.0009118819656
8	0.80000000	0.00045008	0.0003354626279
9	0.90000000	0.00016516	0.0001234098041
10	1.00000000	0.00004540	0.00004539992976

Results for h=0.05:

I	X(I)	W(I)	Y_actual
0	0.00000000	2.00000000	1
1	0.50000000	1.67352715	0.6065306597
2	0.10000000	1.40760181	0.3678794412
3	0.15000000	1.19277117	0.2231301601
4	0.20000000	1.02113909	O. 1353352832
5	0.25000000	0.88611213	0.08208499862
6	0.30000000	0.78218714	0.04978706837
7	0.35000000	0.70477347	0.03019738342
8	0.40000000	0.65004438	0.01831563889
9	0.45000000	0.61481258	0.01110899654
10	0.50000000	0.59642621	0.006737946999
11	0.55000000	0.59268178	0.004086771438
12	0.60000000	0.60175137	0.002478752177
13	0.65000000	0.62212172	0.001503439193
14	0.70000000	0.65254328	0.0009118819656
15	0.75000000	0.69198762	0.0005530843701
16	0.80000000	0.73961176	0.0003354626279
17	0.85000000	0.79472840	0.0002034683690
18	0.90000000	0.85678099	0.0001234098041
19	0.95000000	0.92532290	0.00007485182989
20	1.00000000	1.00000000	0.00004539992976

Numerical Analysis 2 – Amirhossein Mahmoudi – HW3

11.4

4.c

4. Use the Nonlinear Finite-Difference Algorithm with $TOL = 10^{-4}$ to approximate the solution to the following boundary-value problems. The actual solution is given for comparison to your results.

a. $y'' = y^3 - yy'$, $1 \le x \le 2$, $y(1) = \frac{1}{2}$, $y(2) = \frac{1}{3}$; use $h = 0.1$; actual solution $y(x) = (x+1)^{-1}$.

b. $y'' = 2y^3 - 6y - 2x^3$, $1 \le x \le 2$, $y(1) = 2$, $y(2) = \frac{5}{2}$; use $h = 0.1$; actual solution $y(x) = x + x^{-1}$.

c. $y'' = y' + 2(y - \ln x)^3 - x^{-1}$, $2 \le x \le 3$, $y(2) = \frac{1}{2} + \ln 2$, $y(3) = \frac{1}{3} + \ln 3$; use $h = 0.1$; actual solution $y(x) = x^{-1} + \ln x$.

d. $y'' = (y')^2 x^{-3} - 9y^2 x^{-5} + 4x$, $1 \le x \le 2$, $y(1) = 0$, $y(2) = \ln 256$; use $h = 0.05$; actual solution $y(x) = x^3 \ln x$.

This is of the form $y'' = f(x, y, y')$ for $a \le x \le b$ with $y(a) = \alpha$ and $y(b) = \beta$

Chose N

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

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

$x_i = a + ih$ for $i = 0,1,\ldots,N+1$

If $y(x_i) = w_i$, $y(x_{i+1}) = w_{i+1}$, $y(x_{i-1}) = w_{i-1}$:

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

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

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

For each i=1,2,…,N

The corresponding N*N system of equations is given by:

$$2w_1 - w_2 + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) - \alpha = 0$$

$$-w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) = 0$$

$$-w_2 + 2w_3 - w_4 + h^2 f\left(x_3, w_3, \frac{w_4 - w_2}{2h}\right) = 0$$

…..

Numerical Analysis 2 – Amirhossein Mahmoudi – HW3

$$-w_{n-2} + 2w_{n-1} - w_n + h^2 f\left(x_{n-1}, w_{n-1}, \frac{w_n - w_{n-2}}{2h}\right) = 0$$

$$-w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta = 0$$

I	X(I)	W(I)	Y_actual
0	2.00000000	1.19314718	1.193147181
1	2.10000000	1.21813602	1218127821
2	2.20000000	1.24301611	1243002815
3	2.30000000	1.26770778	1267691732
4	2.40000000	1.29215241	1292135404
5	2.50000000	1.31630729	1.316290732
6	2.60000000	1.34014180	1.340126830
7	2.70000000	1.36363457	1.363622143
8	2.80000000	1.38677131	1386762274
9	2.90000000	1.40954321	1.409538323
10	3.00000000	1.43194562	1.431945622

Convergence in 2 iterations