Assignment 4 Key for JDEP 384H

Thomas Shores
Department of Mathematics
University of Nebraska
Spring 2007

Points: 45
Due: April 5

1. (10 pts) In Matlab create an anonymous function $f(x)$ with the formula
`f = @(x) x.^4 - 4*x.^3 + 6*x.^2 - 4*x + 1`

(a) What function does this represent and what is its derivative?

(b) Plot this function on the interval $[1 - dx, 1 + dx]$ in steps of $dx/100$ with $dx = 0.001$ and also $dx = 0.0001$. Are the graphs reasonable? Explain.

(c) Find the value of k such that using $h = 10^{-k}$ gives the best approximation to $f'(0)$ using forward differences and calculate the relative error of the approximation.

(d) Same question as (c) using centered differences.

Solution.

(a) The function represented is

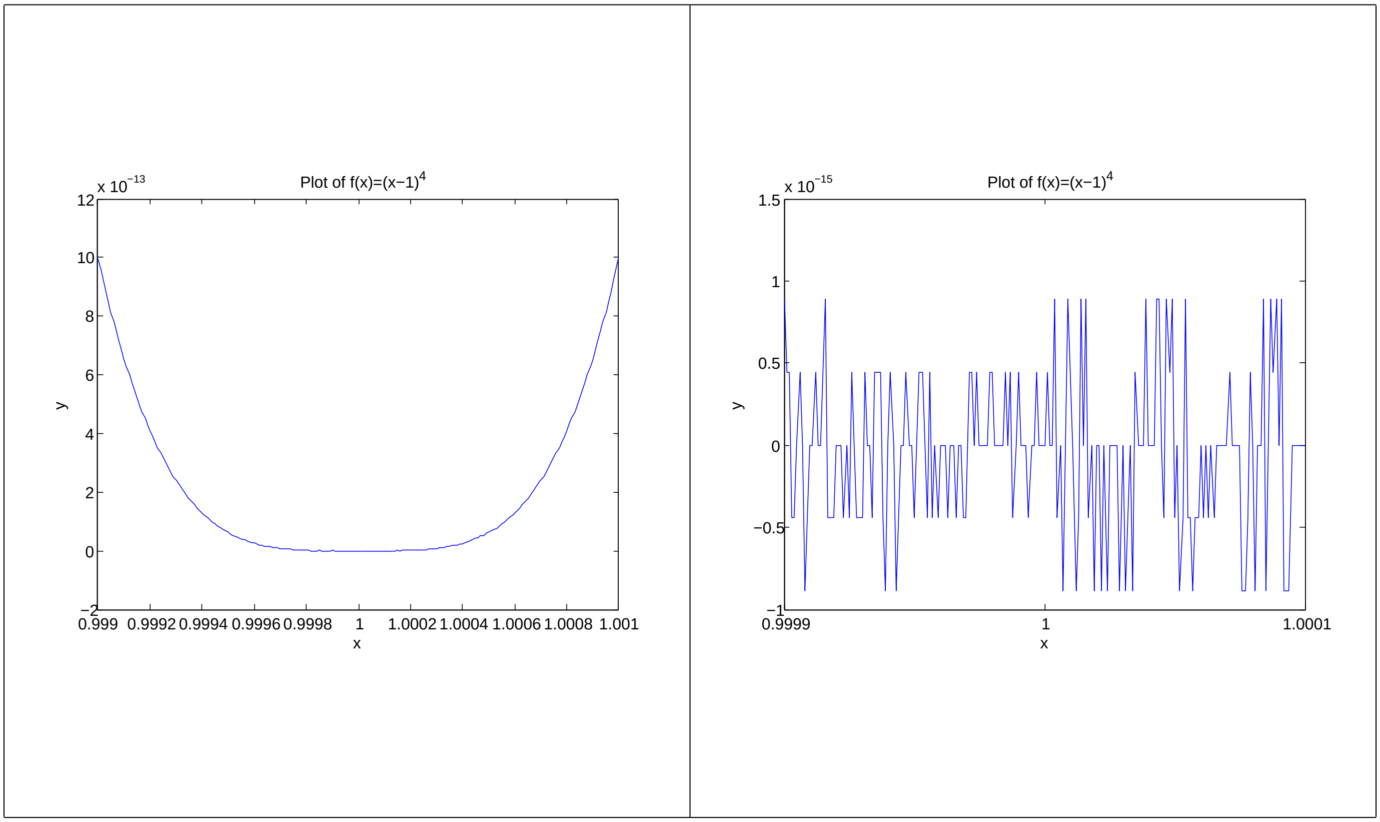
$$f(x) = x^4 - 4x^3 + 6x^2 - 4x + 1 = (x-1)^4.$$

Hence

$$f'(x) = 4x^3 - 12x^2 + 12x - 4 = 4(x-1)^3.$$

(b) The plots are generated by the commands

```
dx = 0.001;
x = [1-dx:dx/100:1+dx];
f = @(x) x.^4 - 4*x.^3 + 6*x.^2 -4*x +1;
figure
plot(x,f(x))
xlabel('x')
ylabel('y')
title('Plot of f(x)=(x-1)^4')
dx = 0.0001;
x = [1-dx:dx/100:1+dx];
figure
plot(x,f(x))
xlabel('x')
ylabel('y')
title('Plot of f(x)=(x-1)^4')
```



The first graph is reasonable, since $f(1) = 0$ and f is smooth. The second is not reasonable, since it should be like the first, only flatter since we are looking at a smaller interval about zero.

(c) We calculate for forward differences (one could do this directly, by inspection):

```
dx = 10.^(-(1:15)); % test steps
fp0 = (f(dx)-f(0))./dx;
error = abs(fp0 - (-4));
ndx = find(error == min(error))ndx =
9
error(ndx)
ans =
2.1054e-009
```

Thus, the best result is with $h = 10^{-9}$ and the error is about $2.1 \cdot 10^{-9}$.

(d) We calculate for centered differences:

```
fp0 = (f(dx)-f(-dx))./(2*dx);
error = abs(fp0 - (-4));
ndx = find(error == min(error))
error(ndx)
ndx =
6
ans =
4.0004e-012
```

Thus, the best result is with $h = 10^{-6}$ and the error is about $4 \cdot 10^{-12}$, which is definitely better than forward differences.

2. (10 pts) Consider the system

$$\begin{aligned} 4x_1 + 3x_2 + 3x_3 &= 5 \\ 3x_1 + 4x_2 + 3x_3 &= 3 \\ 3x_1 + 3x_2 + 4x_3 &= 2. \end{aligned}$$

(a) Find the solution to this system.
(b) Exhibit the iteration matrices for the Jacobi and Gauss-Seidel methods.
(c) Find the eigenvalues of matrix of (b) and indicate which method will converge.
(d) Find the number of iterations of the convergent method that are required to reduce the absolute error to below 0.01, starting with the zero solution.

Solution.

(a) We use Matlab to generate a solution:

```
A = [4,3,3;3,4,3;3,3,4]
A =
4 3 3
3 4 3
3 3 4
b = [5,3,2]'
b =
5
3
2
xtrue = A\b
xtrue =
2.0000
-0.0000
-1.0000
% (c)
%(d)
```

(b) We use Matlab to generate these matrices

```
D = diag(diag(A));
U = triu(A,1);
L = tril(A,-1);
GJacobi = inv(D)*(-L-U)
GJacobi =
0 -0.7500 -0.7500
-0.7500 0 -0.7500
-0.7500 -0.7500 0
GGaussSeidel = inv(D+L)*(-U)
GGaussSeidel =
0 -0.7500 -0.7500
0 0.5625 -0.1875
0 0.1406 0.7031
```

(c) Calculate the eigenvalues and the largest of these in absolute value to get the spectral radius of the iteration matrices.

```
eGJ = eig(GJacobi)
eGJ =
-1.5000
```

```
0.7500
0.7500
rhoGJacobi = max(abs(eGJ))
rhoGJacobi =
1.5000
eGG = eig(GGaussSeidel)
eGG =
0
0.6328 + 0.1464i
0.6328 - 0.1464i
rhoGGaussSeidel = max(abs(eGG))
rhoGGaussSeidel =
0.6495
```

From this we see that the spectral radius of the Jacobi iteration matrix is greater than one, so it will not give a convergent algorithm, while the spectral radius of the Gauss-Seidel iteration matrix is 0.6495, so this matrix will yield a convergent algorithm.

(d) This could be checked directly by hand, or use a loop as below.

```
x = zeros(3,1);
Gb = inv(D+L)*b;
for k = 1:1000
x = GGaussSeidel*x + Gb;
if (norm(x-xtrue,inf)<0.01)
break
end
end
disp(k)
12
```

We see that it takes 12 iterations for Gauss-Seidel to converge to an error smaller than 0.01.

3. (7 pts) Let $\mathbf{x} = (x_1, x_2)$ and $F(\mathbf{x}) = \left(x_1^2 - 3x_2^2 + 3, \sin\left(\frac{\pi}{12}x_1x_2\right) + 1\right)$.

(a) Use Matlab's fminsearch function to find a solution to $F(\mathbf{x}) = \mathbf{0}$ as we did in an optimization example calculation.

(b) Assume that $x_1x_2 = -30$, and use this to eliminate x_1 from the first coordinate of F. Use fzero to find a zero of this first coordinate as a function of x_2 and evaluate F at the resulting (x_1, x_2).

Solution.

(a) We do this by minimizing the norm of the vector $F(\mathbf{x})$:

```
F = @(x) [x(1)^2 - 3*x(2)^2 + 3, sin(pi/12*x(1)*x(2)) + 1];
f = @(x) norm(F(x))^2;
fminsearch(f,[0,0])
ans =
-3.0000 2.0000
```

From this we see that F has a zero at $\mathbf{x} = (-3, 2)$.

(b) If we assume that $x_1x_2 = -30$, then we obtain that $x_1 = -30/x_2$. Plug this into the formula for the first coordinate of F to obain

$$g(x_1) = x_1^2 - 3x_2^2 + 3 = (-30/x_2)^2 - 3x_2^2 + 3 = \frac{900}{x_2^2} - 3x_2^2 + 3.$$

Now use fzero as below to find a root x_1 of g, starting with a guess of 1 and then evaluate F at the point $(-30/x_2, x_2)$:

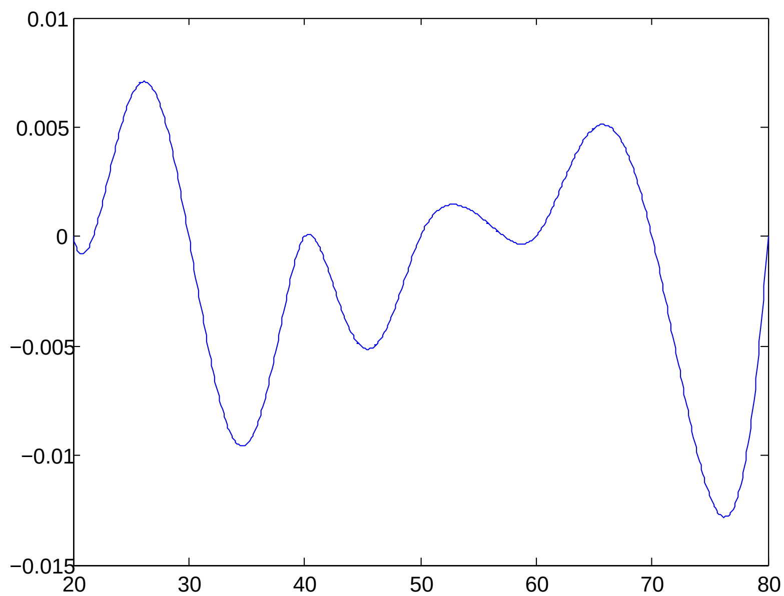
```
g = @(x) 900./(x.^2) -3*x.^2;
x2 = fzero(g,1)
x2 =
4.2223
F([-30/x2,x2])
ans =
1.0e-014 *
-0.7105 0
```

So we deduce that there is a root at $x_2 = 4.1618$ and evaluation gives $F(x_1, x_2) = (3, 0)$.

4. (8 pts) Refer to the European put option of Problem 2 of the take-home midterm. Make a plot of the difference between the true value of the option on the interval $20 \le S \le 80$ at $t = 0$ and the approximation to this curve that you obtain by creating a cubic spline with Matlab that interpolates the curve at the points $S = 20, 30, 40, 50, 60, 70, 80$. What is the largest error on this interval?

Solution. Here is the script that generates the maximum error of 0.0128 and error graph below.

```
% from take-home midterm
t = 0; % current time
K = 50; % strike price
r = 0.1; % current risk-free interest rate
T = 6/12 % expiry
sigma = 0.4; % volatility
S = 20:.1:80; % interval of interest
D0 = 0; % no dividends
EP = bseurput(S,K,r,T,t,sigma,D0);
Snodes = 20:10:80; % get the spline node values
EPnodes = bseurput(Snodes,K,r,T,t,sigma,D0);
spln = spline(Snodes,EPnodes);
EPspline = ppval(spln,S);
plot(S,EP-EPspline); % plot European put
% worst error
max(abs(EP-EPspline))
ans =
0.0128
```



5. (10 pts) Let $f(x) = e^{-x^2/2}/\sqrt{2 * \pi}$, the probability density function for the standard normal distribution. Let $I = \int_0^2 f(x)\,dx$.

(a) Compute an accurate value to I by using statistical functions in Matlab and confirm it with Matlab's quad function.

(b) Approximate the integral using 100 function evaluations and the trapezoidal method.

(c) Approximate the integral using Gaussian quadrature and 8 evaluations.

(d) Approximate I using 1000 evaluations and the hit or miss Monte Carlo method.

(e) In each of (b)-(d), calculate cost (number of function evaluations needed) per correct digit of the answer.

Solution.

(a) First we define the function, which is the pdf for the standard normal distribution, and find the value of the integral using statistical functions.

```
f = @(x) exp((-x.^2)/2)/sqrt(2*pi);
format long % let's see all the digits
Itrue = stdn_cdf(2) - stdn_cdf(0) % using stat functions
Itrue =
0.477249868051821
quad = quad(f,0,2) % using quad
Iquad =
0.47724985697080
```

It appears that quad has 9 correct digits, assuming the statistic functions are accurate.

(b) Here is the trapezoidal calculation:

```
x = linspace(0,2,100); % evaluation nodes
```

```
h = x(2) - x(1); % stepsize
n = length(x);
y = f(x);
Itrap = 0.5*h*sum(f(x(1:n-1))+f(x(2:n)))
Itrap =
0.477246195596682
```

Thus the trapezoidal method has 5 correct digits.

(c) Gaussian quadrature with 8 nodes:

```
IGauss = GaussInt(f,[0,2],8)
IGauss =
0.47724986805256
```

Thus Gaussian quadrature has 11 correct digits.

(d) The hit or miss method:

```
rand('seed',0)
A = 2/sqrt(2*pi); % area of box containing curve
N = 1000;
X = (2-0)*rand(N,1); % scale to width of box
Y = 1/sqrt(2*pi)*rand(N,1); % scale to height of box
hits = sum(Y <= f(X));
Ihitmiss = A*(hits/N)
Ihitmiss =
0.47793285192092
```

Thus the hit or miss method has 3 correct digits.

(e) Evaluations per correct digit are as follows:

Trapezoidal: $100/5 = 20$ evaluations per digit.

Gaussian quadrature: $8/11 \approx 0.7272$ evaluations per digit.

Hit or miss: $1000/3 \approx 333$ evaluations per digit.