

Ceng 375 Numerical Computing
Midterm
Nov 14, 2011 15.40–17.30
Good Luck!

1. (30 pts) Consider the difference approximation for derivative of a function;

$$f'_n = \frac{-f_{n+2} + 4f_{n+1} - 3f_n}{2h}$$

where f_n means $f(x)$ and f_{n+1} means $f(x + h)$

- i (15 pts) Use this formula to approximate the derivative of $f(x) = \cos(x)$ at $x = 0$ using step sizes of $h = 0.10$ and $h = 0.20$.
- ii (15 pts) Make an error analysis by finding the difference with the exact value at each step size, h . Estimate the order of error ($O(h^2)$) by the ratio of these errors.

- i) For $f(x) = \cos(x)$, we then have:

$$h = 0.2 : f'(0) = \frac{-\cos(2(0.2)) + 4\cos(0.2) - 3\cos(0)}{2(0.2)}$$

```
>> a=(-1*cos(0.4)+4*cos(0.2)-3*cos(0))/0.4  
a = -0.0020
```

$$h = 0.1 : f'(0) = \frac{-\cos(2(0.1)) + 4\cos(0.1) - 3\cos(0)}{2(0.1)}$$

```
>> b=(-1*cos(0.2)+4*cos(0.1)-3*cos(0))/0.2  
b = -2.4958e-04
```

- ii) The exact value is: $f'(0) = \sin(0) = 0$. Since we know the actual value, then the actual errors are:

$$h = 0.2 : \text{Error}(0.2) = (0) - (-0.0020) = 0.0020$$

$$h = 0.1 : \text{Error}(0.1) = (0) - (-2.4958e - 04) = 2.4958e - 04$$

The ratio of these errors is:

$$\frac{\text{Error}(0.2)}{\text{Error}(0.1)} = \frac{0.0020}{2.4958e - 04} = 7.9601$$

Cutting the step size in half should reduce the error by a factor of 8. In this case, the reduction is by a factor of 7.9601, which is considering round-off, acceptably close to 8. Therefore we expect the error to be $O(h^4)$.

2. (30 pts) Consider the solution to $f(x) = 0.5$ where $f(x) = x^3$. Choosing initial guesses of $x_a = 0$ and $x_b = 1$,

i (5 pts) Describe the general working of a bracketing method. What are the assumptions for this family of methods? Are these assumptions satisfied for $f(x)$?

- The function $f(x)$ changes signs at the two x -values and, if $f(x)$ is continuous, there must be at least one root between the values.
- The test to see that $f(x)$ does change sign between points a and b is to see if $f(a) * f(b) < 0$

```
>> syms x;
>> fx='sqrt(x)-cos(x)';
fx = sqrt(x)-cos(x)
>> subs(fx,0.0)
ans = -1
>> subs(fx,1.0)
ans = 0.45969769413186
```

ii (10 pts) Write down an expression to show how the error ε_n in the bisection method decreases with subsequent iterations. Find the required number of iterations when the error after n iterations is 10^{-4}

iii (15 pts) Using the bisection method, determine the solution to four decimal places by filling the table below. Does the number of iterations this took agree with the predicted number in previous item?

i	x_1	x_2	x_3	$f(x_3)$	$ x_3 - x_{exact} $
1	0.00000	1.00000	0.50000	0.12500	0.29370
2	0.50000	1.00000	0.75000	0.42188	0.04370
3	0.75000	1.00000	0.87500	0.66992	0.08130
4	0.75000	0.87500	0.81250	0.53638	0.01880
5	0.75000	0.81250	0.78125	0.47840	0.01245

It is said *an expression*, not an explanation of formalism or methodology;

$$\text{error after } n \text{ iterations} < \left| \frac{(b-a)}{2^n} \right|$$

where $b - a$ is the interval ends and n is the number of iterations. In the second part, it is given that “determine the solution to four decimal places” so;

$$\text{error after } n \text{ iterations} = 10^{-4} \leq \left| \frac{(1-0)}{2^n} \right|$$

$$10^4 = 2^n \rightarrow \ln(10^4) = \ln(2^n) \rightarrow \ln(10^4) = n \rightarrow n = 13.288 \rightarrow n \simeq 13$$

The exact x value can be found as $0.5^{1/3} = 0.79370 = x_{exact}$;

i	x_1	x_2	x_3	$f(x_3)$	$ x_3 - x_{exact} $
1	0	1	0.5	0.12500	0.29370
2	0.5	1	0.75	0.42188	0.04370
3	0.75	1	0.875	0.66992	0.08130
4	0.75	0.875	0.8125	0.53638	0.01880
5	0.75	0.8125	0.78125	0.4784	0.01245
6	0.78125	0.8125	0.79688	0.50602	0.00318
7	0.78125	0.79688	0.789078	0.49130	0.00463
8	0.78907	0.79688	0.79298	0.49164	0.00072
9	0.79298	0.79688	0.79493	0.50233	0.00123
10	0.79298	0.79493	0.79396	0.50049	0.00026
11	0.79298	0.79396	0.79347	0.49956	0.00023
12	0.79347	0.79396	0.79372	0.50004	0.00002

The number found by the expression above is $n \leq 13$, so the number of iterations took agree with the predicted number.

3. (25 pts) Consider the function $f(x)$, on $[0, 1]$, defined by

$$f(x) = \sqrt{x} - \cos(x)$$

i (10 pts) Describe how the secant method determine a smaller sub-interval containing a root.

To determine a root of $f(x) = 0$, given two values, x_0 and x_1 , that are near the root,
 If $|f(x_0)| < |f(x_1)|$ Then
 Swap x_0 with x_1
 Repeat
 Set $x_2 = x_1 - f(x_1) * \frac{(x_0 - x_1)}{f(x_0) - f(x_1)}$
 Set $x_0 = x_1$
 Set $x_1 = x_2$
 Until $|f(x_2)| < \textit{tolerance value}$

ii (15 pts) Apply the secant method to $f(x)$ twice.

```
>> x0=1.0;
>> x1=0.0;
>> syms x;
fx='sqrt(x)-cos(x)';
>> subs(fx,x0)
ans = 0.45969769413186
>> subs(fx,x1)
ans = -1
>> x2=x1-subst(fx,x1)*((x0-x1)/(subs(fx,x0)-subs(fx,x1)))
x2 = 0.68507335732605
>> x0=x1
x0 = 0
>> x1=x2
x1 = 0.68507335732605
>> x2=x1-subst(fx,x1)*((x0-x1)/(subs(fx,x0)-subs(fx,x1)))
x2 = 0.65039498012836
>> solve('sqrt(x)-cos(x)')
ans = .64171437087288265839856530031652
```

4. (25 pts) Find the LU factorization of

$$A = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 2 & 5 & 2 & 2 \\ -1 & -3 & -3 & 5 \\ 1 & 3 & 2 & 2 \end{bmatrix}$$

by Gaussian elimination (without pivoting). Clearly show how you get the entries of L and U .

$$\begin{bmatrix} 1 & 3 & 1 & 1 \\ 2 & 5 & 2 & 2 \\ -1 & -3 & -3 & 5 \\ 1 & 3 & 2 & 2 \end{bmatrix}, \begin{array}{l} R_2 - (2/1)R_1 \rightarrow \\ R_3 - (-1/1)R_1 \rightarrow \\ R_4 - (1/1)R_1 \rightarrow \end{array} \begin{bmatrix} 1 & 3 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

$$R_4 - (1/-2)R_3 \rightarrow \begin{bmatrix} 1 & 3 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & 0 & 4 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -0.5 & 1 \end{bmatrix}, \text{ and } L*U = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 2 & 5 & 2 & 2 \\ -1 & -3 & -3 & 5 \\ 1 & 3 & 2 & 2 \end{bmatrix}$$

```
>> L=[1 0 0 0 ; 2 1 0 0; -1 0 1 0; 1 0 -0.5 1]
```

```
L =
```

```

1.0000000000000000      0      0      0
2.0000000000000000  1.0000000000000000      0      0
-1.0000000000000000      0  1.0000000000000000      0
1.0000000000000000      0 -0.5000000000000000  1.0000000000000000
```

```
>> U=[1 3 1 1; 0 -1 0 0; 0 0 -2 6; 0 0 0 4]
```

```
U =
```

```

1      3      1      1
0     -1      0      0
0      0     -2      6
0      0      0      4
```

```
>> L*U
```

```
ans =
```

```

1      3      1      1
2      5      2      2
-1     -3     -3      5
1      3      2      2
```