# Semester 2 Tutorial 1

Submit your solution sheet by the submission deadline specified on Blackboard.

1. How many iteration steps are required when approximating the solution to  using the Bisection method in the interval [a, b] = [2, 2.5] and the termination condition is   
   a) , b) and c) ? Which of these is the most appropriate termination condition?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i, | xa, | xb, | xc, | fa, | fb, | fc, | |xa-xc|, | |xa-xc|/xc |
| 1 | 2.000000 | 2.500000 | 2.250000 | -17.000000 | 48.656250 | 8.665039 | 0.250000 | 0.111111 |
| 2 | 2.000000 | 2.250000 | 2.125000 | -17.000000 | 8.665039 | -5.669403 | 0.125000 | 0.058823 |
| 3 | 2.125000 | 2.250000 | 2.187500 | -5.669403 | 8.665039 | 1.088763 | 0.062500 | 0.028571 |
| 4 | 2.125000 | 2.187500 | 2.156250 | -5.669403 | 1.088763 | -2.388233 | 0.031250 | 0.014493 |
| 5 | 2.156250 | 2.187500 | 2.171875 | -2.388233 | 1.088763 | -0.674747 | 0.015625 | 0.007194 |
| 6 | 2.171875 | 2.187500 | 2.179688 | -0.674747 | 1.088763 | 0.200687 | 0.007812 | 0.003584 |
| 7 | 2.171875 | 2.179688 | 2.175781 | -0.674747 | 0.200687 | -0.238601 | 0.003906 | 0.001795 |
| 8 | 2.175781 | 2.179688 | 2.177734 | -0.238601 | 0.200687 | -0.019351 | 0.001953 | 0.000897 |
| 9 | 2.177734 | 2.179688 | 2.178711 | -0.019351 | 0.200687 | 0.090569 | 0.000976 | 0.000448 |
| 10 | 2.177734 | 2.178711 | 2.178223 | -0.019351 | 0.090569 | 0.035584 | 0.000488 | 0.000224 |
| 11 | 2.177734 | 2.178223 | 2.177979 | -0.019351 | 0.035584 | 0.008110 | 0.000244 | 0.000112 |
| 12 | 2.177734 | 2.177979 | 2.177856 | -0.019351 | 0.008110 | -0.005622 | 0.000122 | 0.000056 |
| 13 | 2.177856 | 2.177979 | 2.177917 | -0.005622 | 0.008110 | 0.001243 | 0.000061 | 0.000028 |
| 14 | 2.177856 | 2.177917 | 2.177887 | -0.005622 | 0.001243 | -0.002189 | 0.000030 | 0.000014 |

1. Evaluate the following polynomial at *x* = 1.07using three-digit rounding,

2.75*x*3 – 2.95*x*2 + 3.16*x* – 4.67

Find the relative error of your result then evaluate the polynomial in nested form and comment on the new result.

Exact result -1.297386749999999  
Rounded result -1.3, Error 0.026

Horner’s method -1.28, Error 0.017

1. What is the error bound when an approximation of *f*(*x*) = (1 + *x*)-2 at x = 1.05 is obtained by the third-degree Taylor polynomial for *f*(*x*) = (1 + *x*)-2 about *x*0 = 1.00 Compare the error bound to the exact value of *f*(1.05).  
   *f*(1.05) = 0.237953599048186



|*f*(1.5)-*p*3(1.05)| ≈ 4.74e-7

1. Let . Calculate the error bound given by  for both the linear and the second-degree interpolating Lagrange polynomial at *x* = 3/5π.

|  |  |
| --- | --- |
| *x*i | *f*(*x*i) |
|  | 0.555360367269796 |
|  | 1.570796326794897 |
|  | 1.666081101809387 |

*f*(*x*) =1.792699298844934

*p*1(*x*)= 3\*0.555360367269796\*0.1 + 7\*1.666081101809387\*0.1= 1.332864881447510

*p*2(*x*)= 0.12\*0.555360367269796 + 0.84\*1.570796326794897+0.28\*1.666081101809387 = 1.719328378941966



F:\L5_NA\tutorials\S2T1Q4a.emf

Error bound *R*1 ≤ 3.08 π2 5.25e-2/2 ≈ 0.798  
compared to exact error of |*f*(3π/5) - *p*1| ≈ 0.460

F:\L5_NA\tutorials\S2T1Q4b.emf

Error bound *R*2≤ 3.242 π3 5.25e-3/6 ≈ 0.0880  
compared to exact error of |*f*(3π/5) - *p*2| ≈ 0.0734

# Semester 2 Tutorial 2

1. Let  and h = 0.2 and 0.02. Find an estimate for f'(x) at x0 = 1 using both forward and backward differences. Why have the slopes got different signs?

|  |  |
| --- | --- |
| *x* | *f*(*x*) |
| 0.8 | 2.7819 |
| 0.98 | 2.7188 |
| 1 | 2.7183 |
| 1.02 | 2.7188 |
| 1.2 | 2.7668 |



Because the function has a minimum at x = 1

2. Find the error bound for each of the four estimates and compare with the actual error.



h=-0.2, ξ=0.8: error bound 0.45 > 0.32

h=-0.02, ξ=0.98: error bound 0.028 >0.0276

h=0.02, ξ=1: error bound 0.02718 > 0.0268

h=0.2, ξ=1: error bound 0.2718 > 0.2424

1. Use the central difference approximation to find the estimate for f'(x). Is it a better or worse approximation? Find a graphical explanation.



(0.2424-0.3182)/2=-0.0379

(0.0268-0.0276)/2=-0.0004

Both are better because the approximation averages the two slopes, therefore it is in essence a higher order approximation with *x*0 in the centre!

4.

You have information about the following function values:

|  |  |
| --- | --- |
| *x* | *f*(*x*) |
| -0.3  -0.1  0.1 | -0.0431  -0.08993  0.11007 |

Find an estimate for *f* '(-0.3) by substituting the values into the *n*+1 formula.



Remember to differentiate *L*(*xj*)

*L*0'(*x*)=(2*x* - *x*1 - *x*2)/(*x*0-*x*1)/(*x*0-*x*2), *L*1'(*x*)=(2*x* - *x*0 - *x*2)/(*x*1-*x*0)/(*x*1-*x*2), *L*2'(*x*)=(2*x* - *x*0 – *x*1)/(*x*2-*x*0)/(*x*2-*x*1)

*p*2 = -0.6/(-0.2)/(-0.4)\*(-0.0431) – 0.4/0.2/(–0.2)\*(–0.08993)–0.2/0.4/0.2\*0.11007

=0.32325-0.8993-0.275175=-0.851225

# Solution Tutorial 3

1. Use either three-point formula (1) or (2) to determine approximations that will complete the following tables.

|  |  |  |
| --- | --- | --- |
| *x* | *f*(*x*) | *f'*(*x*) |
| -0.3  -0.1  0.1  0.3 | -0.20431  -0.08993  0.11007  0.39569 | 0.35785 (eq.1) 0.78595 (eq.2)  1.21405 (eq. 2)  1.64215(eq.1) |



|  |  |  |
| --- | --- | --- |
| *x* | *f*(*x*) | *f'*(*x*) |
| 1.1  1.2  1.3  1.4 | 0.48603  0.86160  1.59751  3.76155 | 1.954  5.5574 14.49975 28.78105 |





1. Consider the following table of data and use the five-point formula to approximate *f*’(0.6)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| *f*(*x*) | 0.9798652 | 0.9177710 | 0.8080348 | 0.6386093 | 0.3843735 |



Answer: -0.6824174583Consider the data from (2) and use all appropriate formulae to approximate *f’*(0.4)

|  |  |
| --- | --- |
|  | -0.548681 |
|  | -0.310471 |
| 3-point (2) | -0.429576 |
| 3-point(1) | -0.667786 |

In order of most accurate to least accurate: 3-point (2), 3-point (1), 2-point backwards, 2-point forwards. Accuracy estimated from difference between most accurate estimate and less accurate estimate. Better to draw function values and estimate possible accuracy from this.

1. Let *f*(*x*) = cos (π*x*)*.*
   1. Use the five-point formula and the values of *f*(*x*) at *x* = 0, 0.25, 0.75, and 1.0 to approximate *f*’(0.5).
   2. Find a bound for the error.
2. f’(0.5) = -3.10456949966159

Solution Tutorial 4

1. Suppose the following data has been experimentally collected.

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | 1.00 | 1.01 | 1.02 |
| *f*(*x*) | 1.27 | 1.32 | 1.38 |

Approximate *f*’(1.005) and *f*’(1.015) using first order three-point formula (2). Is it possible to find a bound for the error of that approximation?

1st order 3-point formula



error bound can't be provided. 1) function unknown, therefore second derivative unknown too. 2) *x ≠ xk*

1. Let Approximate *f*’(1.05) using *h* = 0.05 and *h* = 0.01 in first-order three-point formula (2) with the following data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 1.0 | 1.04 | 1.05 | 1.06 | 1.10 |
| *f*(*x*) | 1.6829420 | 1.7732994 | 1.7960257 | 1.8188014 | 1.9103448 |

Use the Matlab code provided in the lecture to compare these approximations with the analytical result. Provide the code, the Matlab output and the result with your answer.

Matlab code:

diff('sin(x)\*2^x','x')  
ans =  
 cos(x)\*2^x+sin(x)\*2^x\*log(2)

x=1.05;eval(ans)

ans =  
 2.2751

Approximation 1st order derivative



1. Consider the data below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| f(x) | 0.9798652 | 0.9177710 | 0.8080348 | 0.6386093 | 0.3843735 |

In choosing *h* appropriately

1. use all appropriate higher-order formulae to approximate *f* ’’(0.4)
2. use all appropriate higher-order formulae to approximate *f* ’’(0.6)



1. Using the data from Question 1
2. Approximate *f ’’*(1.01), using 2nd degree Three-point formula and the results of Question 1.
3. Suppose the measured data set is accurate to within ±0.0005. Find the error due bound to data accuracy.



Notes: In a) it is reassuring that two fundamentally different methods (1st based on Taylor polynomials, 2nd based on Lagrange polynomials and applied twice) yields exactly the same result.

In b) we do not know the function and can therefore not estimate the truncation error bound.

Solution Tutorial 5

1. The table below describes a car travelling on a straight road. Use the following times and positions at those times, and an appropriate approximation formula to predict the speed at each time listed. Specify which method you have chosen and give a reason for your choice.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time | 0 | 3 | 5 | 8 | 10 | 13 |
| Distance | 0 | 225 | 383 | 623 | 742 | 993 |

*For application of 3-point formulae the table needs to be split into two with equal spacing*

|  |  |  |  |
| --- | --- | --- | --- |
| Time | 0 | 5 | 10 |
| Distance | 0 | 383 | 742 |

|  |  |  |  |
| --- | --- | --- | --- |
| Time | 3 | 8 | 13 |
| Distance | 225 | 623 | 993 |

*Apply second formula to x = 5 and 8 to get a more accurate estimate. And use forward approximation (3-point one) to points that have only one adjacent value. First order approximation could also be explored for the latter to possibly achieve increased accuracy.*

*3-point 2:*

*3-point 1:*

Therefore the table can be written as

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Time | 0 | 3 | 5 | 8 | 10 | 13 |
| Distance | 0 | 225 | 383 | 623 | 742 | 993 |
| Speed (approx.) | 79 | 82.4 | 74.2 | 76.8 | 69.4 | 71.2 |

1. Consider the function



Show that *e*(*h*) has a minimum at . Then evaluate *h* for M being a bound foron [0.8, 1.0] and ε = 5 x 10-4.

*Set first derivative of e(h) =0 then rearrange re h*

*Find second derivative at h=0 and confirm that this is a minimum*

*M=0.6967067: error bound = 0.1291*

1. With *x*1 = *x*0 + *h*, and *x*2 = *x*0 + 3*h* use the equation



to derive an approximation to *f’* (*x*0).



1. Perform the following computations (i) exactly, (ii) using three-digit chopping arithmetic,   
   (iii) using three-digit rounding arithmetic. Then determine absolute and relative errors.
   1. 14.1 + 0.0981
   2. 0.0218 x 179
   3. (164 + 0.913) – (143 + 21)
   4. (164 – 143) + (0.913 – 21)

|  |  |  |  |
| --- | --- | --- | --- |
| Task | exact | 3-digit chopping | 3-difit rounding |
| 14.1+0.0981 | 0.141981x102 | 0.141x102 | 0.142x102 |
| 0.0218\*179 | 0.39022x101 | 0.390 | 0.390 |
| (164+0.913)-(143+21) | 0.913 | 0 | 1 |
| (164-143)+(0.913-21) | 0.913 | 1 | 0.9 |

Solution Not a tutorial any longer

1. From calculus we know that 

Show that

*Use the McLaurin series of degree 2 for sin x*



1. What is the rate of convergence of 

*Use the McLaurin series of degree 2 for cos x*



1. The sequence 3-*n* considered in the example of Lecture 15 can be generated by either of the two relations

i)

ii)

a) Compute  for n = 2, ..., 8 using five-digit rounding arithmetic and compare the results to those obtained in the example.  
 and so on

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 1 | 1 |
| 1 | 0.333333 | 0.33333 | 0.33333 |
| 2 | 0.111111 | 0.1111 | 0.11111 |
| 3 | 0.037037 | 0.03702 | 0.03703 |
| 4 | 0.012346 | 0.01233 | 0.01234 |
| 5 | 0.004115 | 0.0041 | 0.00412 |
| 6 | 0.001372 | 0.00136 | 0.00139 |
| 7 | 0.000457 | 0.00045 | 0.0005 |
| 8 | 0.000152 | 0.00014 | 0.00023 |
| 9 | 5.08E-05 | 0.00004 | 0.00018 |
| 10 | 1.69E-05 | 0.00001 | 0.00024 |
| 11 | 5.65E-06 | 0 | 0.00038 |
| 12 | 1.88E-06 | 0 | 0.00063 |
| 13 | 6.27E-07 | 0 | 0.00105 |
| 14 | 2.09E-07 | 0 | 0.00175 |
| 15 | 6.97E-08 | 0 | 0.00292 |
| 16 | 2.32E-08 | 0 | 0.00487 |
| 17 | 7.74E-09 | 0 | 0.00812 |
| 18 | 2.58E-09 | 0 | 0.01353 |
| 19 | 8.6E-10 | 0 | 0.02255 |

b) Compute  for n = 2, ..., 8 using five-digit rounding arithmetic and compare the results to those obtained in the example.

c) Show whether procedures (i) and/or (ii) are stable.

From table (i) stable (ii) not stable, can also be shown using the rationale shown in the lecture.

1. Apply the extrapolation process to determine *N*3(*h*), an approximation to *f'*(*x*0), for the following function and step sizes, using the example on Richardson’s extrapolation from the lecture (Section 5.6.1 in the notes).

a)

b)

|  |  |  |
| --- | --- | --- |
| a) |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| b) | Exact f(x)=1 |  |
|  |  |  |
|  |  |  |
|  |  |  |

Solution Tutorial 6

1. Use trapezoidal rule to approximate the following integrals.

a) b) c) d)

a)

b)

c)

d) 

1. For the approximations in Question 1 compare the approximation to the actual value and find a bound for the error in each case if possible.

a)

b)

c)

d) 

|  |  |
| --- | --- |
| Absolute error | Relative error |
| 0.0397 | 10.3% |
| 0.0116 | 33.3% |
| 0.0856 | 27.9% |
| 0.0461 | 13.3% |

Error bounds



max at ξ = π/4

error = ∞ at x = 0, no error bound defined

max at ξ = 0

max at ξ = 1

1. Use the functions from Question 1 and approximate the integrals with Simpson's rule.

Simpson's rule











1. For the approximations in Question 3 compare the approximation to the actual value and find a bound for the error in each case if possible.

|  |  |
| --- | --- |
| Absolute error | Relative error |
| -4.7\*10-4 | 1.2\*10-3 |
| 2.5\*10-3 | 0.072 |
| 0.001659805553181 | 0.54% |
| 1.18\*10-3 | 3.4\*10-3 |

Error bound



a) 

b) not defined

c) 

d) 

Solution Tutorial 7

1. Use the following table to find an approximation to , using   
   a) the trapezoidal rule with *x*0 = 1.1, *x*1 = 1.5  
   b) Simpson's rule with *x*0 = 1.1, *x*1 = 1.3, and *x*2 = 1.5

|  |  |
| --- | --- |
|  |  |
| 1.1 | 3.0042 |
| 1.3 | 3.6693 |
| 1.5 | 4.4817 |

1. **(10 marks)**
2. **(10 marks)**
3. Exact = exp(1.5)-exp(1.1)=1.477523
4. Use the Newton-Cotes closed formula for *n* = 4 and open formula for *n* = 3 to approximate.

Closed Newton-Cotes formula *h* = 0.5, nodes at [1, 1.5, 2, 2.5, 3]



**(10 marks)**

Open Newton-Cotes formula *h* = 0.4, nodes at [1.4, 1.8, 2.2, 2.6]



**(10 marks)**

1. Find a bound for the error in each case. Compare the approximation obtained with the actual value 0.7668010.

Error bound, close Newton-Cotes



**(10 marks, see comparison above)**

Error bound open Newton-Cotes



4. Approximate the following integrals using Simpson's rule, Simpson's three-eights rule and midpoint rule when *n* =2.

a)  b) 

Does Part (a) or Part (b) give the better approximation?



* + 1. i) nodes [1, 5.5, 10] 

ii) nodes [1, 4, 7, 10] 

iii) nodes [1, 3.25, 5.5, 7.75, 10]

**(10 marks)**

* + 1. i) nodes [1, 3.25, 5.5] [5.5, 7.75, 10]   
       

ii) nodes [1, 2.5, 4, 5.5][5.5, 7, 8.5, 10] 

iii) nodes [1, 2.125, 3.25, 4.375, 5.5][5.5, 6.625, 7.75, 8.875, 10]

answers b) best because composite Simpson’s rule more accurate than simple one.

**(10 marks)**

Solution Tutorial 8:

1. Use the composite trapezoidal rule with indicated values of *n* to approximate the following definite integrals. Compare the approximations to the exact results.

a) , *n* =4 b) , *n* = 4

a) *h* = 0.5, nodes at [1, 1.5, 2, 2.5,]

error of the order 10-2

b) *h* = 0.5, nodes at [0, 0.5, 1, 1.5, 2 ]   
error of the order 10-1

1. Determine the values of *n* and *h* needed to approximate  to within 10-4 using   
   a) Simpson's composite rule

b) the composite trapezoidal rule and

c) the composite midpoint rule

a)



b)



c)



This answer only gives the positive root while the negative one gives the bigger absolute value. Therefore solve using quadratic equation:



1. A particle of mass *m* moving through a fluid is subjected to a viscous resistance *R*, which is a function of the velocity *v*. The relation between *R*, *v* and time *t* is given by   
   

Suppose that for a particular fluid, where *R* is given in Newton and *v* in ms-1. If *m* = 10 kg and *v*(0) = 10 ms-1, approximate the time required for the particle to slow to *v* = 5

a) Use Simpson's rule

b) Use the trapezoidal composite rule with *h* = 0.25

c) Compare these approximations to the actual values.

a) 

b) 

**%% Q2 b)**

**clear all; close all; clc;**

**sum = 10^(-3/2)+5^(-3/2);**

**for j=1:19**

**x=10-j\*0.25**

**sum=sum+2\*x^(-3/2);**

**end**

**intfx=-0.25\*10\*sum/2**

n = 11 for ε ≤ 10-2

c) 

1. Integrate *f*(*x*) = *x*-2 over the interval [0.2, 1] using adaptive quadrature based on composite Simpson’s rule according to the lecture slides. Use a tolerance value of 0.02 and compare the number of function evaluations with that of composite Simpson’s rule.

*S*1[0.2, 1] = 4.948148

*S*2[0.2, 0.6] = 3.51851852

*S*2[0.6, 1] = 0.66851852

*S*1[0.2, 1] - *S*2[0.2, 0.6] - *S*2[0.6, 1] = 0.7611111 smaller value for *h* required for tol = 0.02(\*15)

For right half

*S*2[0.6, 1] - *S*3[0.6, 0.8] - *S*3[0.8, 1] = 0.66851852 – (0.41678488 + 0.25002572) = 0.001708   
smaller than required tolerance of 0.01\*15

*S*2[0.2, 0.6] - *S*3[0.2, 0.4] - *S*3[0.4, 0.6] = 3.51851852-2.52314815-0.83425926 = 0.161111

larger than required tolerance of 0.01\*15

*S*3[0.4, 0.6] - *S*4[0.4, 0.5] - *S*4[0.5, 0.6] = 0.83425926 – 0.50005144 – 0.33334864 = 0.000859

smaller than required tolerance of 0.005\*15

*S*4[0.3, 0.4] - *S*5[0.3, 0.35] - *S*5[0.35, 0.4] = 0.83356954 – 0.47620166 – 0.35714758 = 0.000220

smaller than required tolerance of 0.0025\*15

*S*4[0.2, 0.3] - *S*5[0.2, 0.25] - *S*5[0.25, 0.3] = 1.66851852 - 1.00010288 - 0.66669728 = 0.001718

smaller than required tolerance of 0.0025\*15

(0.41678488 + 0.25002572)+( 0.50005144 + 0.33334864) + (0.47620166 + 0.35714758) + (1.00010288 + 0.66669728) = 4.000360080000000

Composite Simpsons rule only:

|  |  |  |  |
| --- | --- | --- | --- |
| ***n*** | ***hn*** | ***Sn*** | **|*Sn*+1-*Sn*|** |
| 1 | 0.4 | 4.948148 |  |
| 2 | 0.2 | 4.187037 | 0.761111 |
| 3 | 0.1 | 4.024218 | 0.162819 |
| 4 | 0.05 | 4.002164 | 0.022054 |
| 5 | 0.025 | 4.000154 | 0.002010 |

Solution Tutorial 9:

1. Find and for the following vectors:

* x= for an integer k



2. Find the first two iterations of the Jacobi method for the linear system, using



%% Q2 Jacobi

clear all

x1=0;x2=0;x3=0;

for i=1:15

x1(i+1)=0.5\*x2(i)-0.5\*x3(i)-0.5;

x2(i+1)=-x1(i)-3\*x3(i);

x3(i+1)=-3/5\*x1(i)-3/5\*x2(i)+4/5;

end

x1=x1';

x2=x2';

x3=x3';

A=[2 -1 1;3 3 9;3 3 5];

b=[-1;0;4];

x=A\b; % exact solution

|  |  |  |  |
| --- | --- | --- | --- |
| **k** |  |  |  |
| **0** | 0.0000 | 0.0000 | 0.0000 |
| **1** | -0.5000 | 1.3333 | 0.0000 |
| **2** | 0.1667 | 1.8333 | -0.2778 |
| **…** | … | … | … |
| **15** | 0.9968 | 1.9985 | -0.9975 |

exact solution x=(1,2,-1)t

1. The pair of equations*,*

can be rearranged to give

Apply the Jacobi method with and observe the divergence. Then apply the Gauss-Seidel method. Which method diverges more rapidly?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Jacobi method | |  | Gauss-Seidel method | |
| **k** |  |  |  |  |  |
| **0** | 1.01 | 1.01 |  | 1.01 | 1.01 |
| **1** | 1.02 | 0.97 |  | 1.02 | 0.94 |
| **2** | 0.94 | 0.94 |  | 0.88 | 1.36 |
| **3** | 0.88 | 1.18 |  | 1.72 | -1.16 |
| **4** | 1.36 | 1.36 |  | -3.32 | 13.96 |
| **5** | 1.72 | -0.08 |  | 26.92 | -76.76 |
| **6** | -1.16 | -1.16 |  | -154.52 | 467.56 |
| **7** | -3.32 | 7.48 |  | 934.12 | -2798.36 |
| **8** | 13.96 | 13.96 |  | -5597.72 | 16797.16 |
| **9** | 26.92 | -37.88 |  |  |  |
| **10** | -76.76 | -76.76 |  |  |  |
| **11** | -154.52 | 234.28 |  |  |  |
| **12** | 467.56 | 467.56 |  |  |  |
| **13** | 934.12 | -1398.68 |  |  |  |
| **14** | -2798.36 | -2798.36 |  |  |  |
| **15** | -5597.72 | 8399.08 |  |  |  |

exact solution x = (1, 1)t

|  |  |
| --- | --- |
| %% Q3 Jacobi  clear all  x1=1.01;x2=1.01;  for i=1:15  x1(i+1)=-1+2\*x2(i);  x2(i+1)=4-3\*x1(i);  end  x1=x1';  x2=x2';  A=[1 -2;3 1];  b=[-1;4];  x=A\b; % exact solution | %% Q3 Gauss-Seidel  clear all  x1=1.01;x2=1.01;  for i=1:15  x1(i+1)=-1+2\*x2(i);  x2(i+1)=4-3\*x1(i+1);  end  x1=x1';  x2=x2'; |

1. Find the first two iterations of the Gauss-Seidel method for the linear system, using



%% Q4

clear all

x1=0;x2=0;x3=0;

for i=1:30

x1(i+1)=x2(i)+x3(i)+0.375;

x2(i+1)=x1(i+1)+2\*x3(i);

x3(i+1)=-0.5\*x2(i+1);

end

x1=x1';

x2=x2';

x3=x3';

A=[1 -1 -1;1 -1 2;0 2 4];

b=[0.375;0;0];

x=A\b; % exact solution

|  |  |  |  |
| --- | --- | --- | --- |
| **k** |  |  |  |
| **0** | 0 | 0 | 0 |
| **1** | 0.375 | 0.375 | -0.1875 |
| **2** | 0.5625 | 0.1875 | -0.09375 |
| **3** | 0.46875 | 0.28125 | -0.140625 |
| **4** | 0.515625 | 0.234375 | -0.1171875 |
| **5** | 0.492188 | 0.2578125 | -0.12890625 |
| **6** | 0.503906 | 0.24609375 | -0.123046875 |
| **7** | 0.498047 | 0.251953125 | -0.125976563 |
| **8** | 0.500977 | 0.249023438 | -0.124511719 |

exact solution x = (0.5,0.25,-0.125)t

Solution 2 Tutorial 10

1. Using the tic, toc and magic commands in Matlab find out which way of testing for the singularity of a matrix is most efficient. Provide an explanation for your finding.

>> clear all; close all;

>> A=magic(6)

A =

35 1 6 26 19 24

3 32 7 21 23 25

31 9 2 22 27 20

8 28 33 17 10 15

30 5 34 12 14 16

4 36 29 13 18 11

>> tic

>> inv(A)

>> toc

Warning: Matrix is close to singular or badly scaled. Results may

be inaccurate. RCOND = 1.600321e-18.

ans = 1.0e+15 \*

-1.1259 -0.0000 1.1259 1.1259 -0.0000 -1.1259

-1.1259 -0.0000 1.1259 1.1259 -0.0000 -1.1259

0.5629 0.0000 -0.5629 -0.5629 0.0000 0.5629

1.1259 0.0000 -1.1259 -1.1259 0.0000 1.1259

1.1259 0.0000 -1.1259 -1.1259 0.0000 1.1259

-0.5629 0 0.5629 0.5629 -0.0000 -0.5629

Elapsed time is 0.002070 seconds.

>> tic

>> det(A)

>> toc

ans = -2.2999e-09

Elapsed time is 0.001160 seconds.

>> tic

>> rank(A)

>> toc

ans = 5

Elapsed time is 0.001168 seconds.

>> tic

>> lu(A)

>> toc

ans =

35.0000 1.0000 6.0000 26.0000 19.0000 24.0000

0.1143 35.8857 28.3143 10.0286 15.8286 8.2571

0.8571 0.1154 25.5884 -11.4435 -4.1131 -5.5247

0.2286 0.7739 0.3797 7.6416 -5.0305 5.2221

0.0857 0.8893 -0.7306 0.1953 5.2718 10.5435

0.8857 0.2261 -0.3797 -1.0000 0 0.0000

Elapsed time is 0.000730 seconds.

1. Compute the condition number of the following matrices relative to The matrix norm we are using is the maximum row sum:



a)

finding inverse of matrix A manually: first augment A by identity matrix then reduce using Gauss-Jordan method with exact arithmetic



inverse matrix are last two columns

row2\*2, then row1-row2

row2/1.9998e-4

row2/1.0001, then row1-row2



*K*(**A**)=(20000)(3.0001) = 60002 ill-conditioned

b) 



*K*(**A**)=(8.7)(4.7783)≈41.57 reasonably well conditioned

1. Show whether the following system is well-conditioned or ill-conditioned

 



condition number *K*(**A**)= 100, therefore reasonably welll-conditioned

1. Find the dominant eigenvalue and the corresponding eigenvector by the power method



|  |  |  |  |
| --- | --- | --- | --- |
| **m** | x | z | λ |
| **0** | 1.0 | 1.0000 | 11.0000 |
| **1** | 1.0 | 0.3636 | 9.7273 |
| **2** | 1.0 | 0.2150 | 9.4299 |
| **3** | 1.0 | 0.1744 | 9.3489 |
| **4** | 1.0 | 0.1629 | 9.3259 |
| **5** | 1.0 | 0.1596 | 9.3193 |
| **6** | 1.0 | 0.1587 | 9.3174 |
| **7** | 1.0 | 0.1584 | 9.3168 |
| **8** | 1.0 | 0.1583 | 9.3167 |
| **9** | 1.0 | 1.0000 | 9.3166 |
| **m** | u1 | u2 | u3 | λ |
| **0** | 1.0 | 1 | 1 | 8.0000 |
| **1** | 1.0 | 0.875 | 0 | 7.7500 |
| **2** | 1.0 | 0.7097 | 0.1129 | 7.4194 |
| **3** | 1.0 | 0.6674 | 0.0804 | 7.3348 |
| **4** | 1.0 | 0.6476 | 0.08 | 7.2952 |
| **5** | 1.0 | 0.6402 | 0.0778 | 7.2804 |
| **6** | 1.0 | 0.6371 | 0.0772 | 7.2743 |
| **7** | 1.0 | 0.6359 | 0.077 | 7.2718 |
| **8** | 1.0 | 0.6354 | 0.0769 | 7.2708 |
| **9** | 1.0 | 0.6352 | 0.0768 | 7.2704 |
| **10** | 1.0 | 0.6351 | 0.0768 | 7.2703 |
| **11** | 1.0 | 0.6351 | 0.0768 | 7.2702 |