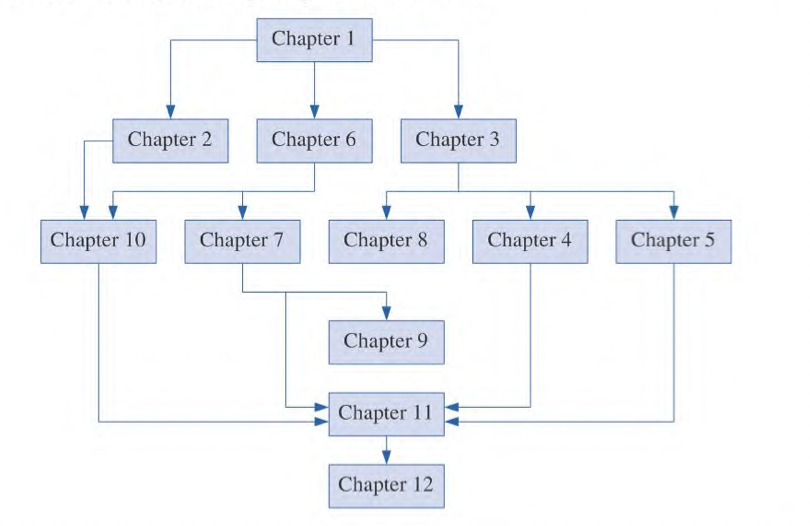
*From the prescribed book, Numerical Analysis (Richard L. Burden, Douglas J. Faires, Annette M. Burden), this would be the recommended order to complete the work in:*



C2: Solutions of Equations in One Variable

C10: Newton’s Method for Nonlinear Systems of Equations

C6: Direct Methods for Solving Linear Systems

C7: Iterative Techniques in Matrix Algebra

C3: Interpolation and polynomial approximation

C4: Numerical Differentiation and Integration

C8: Discrete Least Squares Approximation

**Lesson 0**

C2: Solutions of Equations in One Variable

2.1 The Bisection Method (Root Finding)

ROOT= within interval

* one of the most basic problems of numerical approximation, the root-finding problem.
* This process involves finding a root, or solution, of an equation of the form f, for a given function.
* A root of this equation is also called a zero of the function.
* Based on Intermediate Value Theorem, this is called **Bisection** or **Binary Search Method**

**Bisection method**

1. Get our interval

Check signs

If and have different signs

If and have same sign

2. Valuation step: Do *recurrence relation*

a and b are the same sign:

*Split interval in two. Like the gradient*

3. Repeat until converges

*is the initial value*

*is the last value*

*is the middle point*

*is the number of iterations*

*is the acceptable error*

**Order of Bisection**

[1]

*Current error = constant \* previous error*

[2]

*Previous error = constant \* current error*

Example: <https://www.youtube.com/watch?v=MlP_W-obuNg>

Given

**No. of iterations to find a root.**

*is the initial value*

*is the last value*

*is the middle point*

*is the number of iterations*

*is the acceptable error*

* Depends on acceptable error
* Error decreases by every iteration

Example:

Given

find , the number of iterations it will take to get the acceptable error

Example:

Given that has a root between and

Find the root to decimal place using the bisection method

(Three iterations)

1. Get our interval

*OR use the given starting point(s)*

Let

*To check where the sign changes, sub values into the function and see for which two values of the sign changes*

*Sign changes from negative to positive between and*

Hence, and

1. Valuation step: Do

*Split interval in two. Like the gradient*

If and have different signs

If and have same sign

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Therefore, our root is

Which is within our initial interval

2.2 Fixed-Point Iteration (Root Finding)

*Fixed Point Method*

ROOT= where converges

* Also a problem of finding solutions to fixed point problems
* A fixed point for a function is a number at which the value of the function does not change when the function is applied.

The number is a fixed point for a given function if

Some notes:

T1: if , then will converge

*The error after every iteration is getting smaller*

**How to find the Fixed-Point**

1. Given write in terms of

LHS current guess

RHS *previous guess*

2. Check for if converges

**T1: if , then will converge**

If converges

3. Let initial guess be

*OR use the given starting point(s)*

4. Valuation step: Do *recurrence relation*

5. Repeat until converges

Where

Example: <https://www.youtube.com/watch?v=OLqdJMjzib8>

Given, find where

1. State the recurrence relation

Recurrence relation

1. Given write in terms of

RHS *previous guess*

LHS *current guess*

1. Check for if converges

*T1: if , then will converge*

Root: let

**Wolframalpha**

g(x) = 1+1/x

(look for “root” section)

**Wolframalpha**

g(x) = 1+1/x

converges

3. Let initial guess be

1. Valuation step: Do

|  |  |  |
| --- | --- | --- |
|  | Working out |  |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

2.3 Newton's Method and Its Extensions (Root finding)

(Newton-Raphson Method)

*Is a special case of fixed-point iteration*

ROOT= where converges

- is a lot faster than bisection

**Newton’s method**

1. Pick an close to root of continuous function

2. Derivative

3. Let initial guess be

*OR use the given starting point(s)*

4. Valuation step: Do *recurrence relation*

but

*If the slope is zero, we cannot calculate the next*

4. Repeat until converges

Where

Example: <https://www.youtube.com/watch?v=PIPiv6gn_Ls>

Find the roots of , Three iterations

Given and

1. State the recurrence relation

Recurrence relation

1. Pick an close to root of continuous function
2. Derivative

**Wolframalpha**

f(x) = x^3+2x-2

3. Let initial guess be

*OR use the given starting point(s)*

Given and

Let

4. Valuation step: Do *recurrence relation*

but

*If the slope is zero, we cannot calculate the next*

|  |  |  |
| --- | --- | --- |
|  | Working out |  |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

4. Repeat until converges

Where

Converges around

Example: JUNE 2014 Q1b

*Example showing two different fixed-point algorithms*

*(i.e., fixed point and newtons)*

Assuming that we need to find the roots of

Give two different algorithms for finding the root.

Find derivative

Let

1. Fixed point: Given write in terms of

Fixed point method on

1. Newton’s method

Fixed point method on

Secant method

- Is a variant of Newton’s method

- Newton’s method requires us to know the derivative beforehand

- some functions may be complicated

e.g., and so the derivative won’t be easy to calculate

Newton’s Method

-> tangent (instantaneous rate of change)

Secant Method

-> secant (slope/gradient)

**Newton’s method**

1. Let initial guess be

*OR use the given starting point(s)*

2. Valuation step: Do *recurrence relation*

4. Repeat until converges

Where

**Order of Secant Method**

[1]

*Current error = constant \* previous error*

[2]

*Previous error = constant \* current error*

Example: <https://www.youtube.com/watch?v=_MfjXOLUnyw>

Given

Find root of

1. Let initial guess be

Initial Guess

*Chose two numbers for guesses*

2. Valuation Step, do

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *Sub all into recurrence relation* |  |  | *Sub  into* |
| 0 |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

Regula Falsi method

- Is a variant of Newton’s method

- The Regula Falsi differs from the secant method in that the endpoints have to satisfy the sign test at each iteration. So after each iteration, a decision has to be made of which iterate has to be discarded in order to maintain the bracketing.

Method of False position

*Combination of Secant Method + Bisection Method*

ROOT= where

[Q] What is the main difference between the secant method and the false position method?

**Solution:** The difference between the secant method and the false position method is that in the false position method, beside everything done in the secant method, at each iteration, we check that the two approximations bracket the root.

[Q] Describe the false position (regula falsi) method for solving the equation .

The Regula Falsi (false position) method is an iterative method to solve equations of the form . We start with two initial approximations and and their respective images and . Then set as the x−value of the intersection of the line joining and with the x−axis.

become the new if and have the same sign, or the new if and have the same sign. Then we repeat the process until we find a p such that or the distance between and is less than a given tolerance.

* Based on Intermediate Value Theorem, this is called **Bisection** or **Binary Search Method**

**Regula falsi method**

1. Find root of continuous function

2. Let initial guess be

3. Get our interval

and must have different signs

There must be a zero between these

Check signs

If and have different signs

If and have same sign

4. Valuation step: Do *recurrence relation*

5. Repeat until converges

Example: <https://www.youtube.com/watch?v=pRb3x7zMEHc>

2. Let initial guess be

3. Get our interval

and must have different signs

There must be a zero between these

Check signs

If and have different signs

If and have same sign

4. Valuation step: Do

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 0 |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

If and have different signs

If and have same sign

2.4 Error Analysis for Iterative Methods 78

Remember that:

*Current error = constant \* previous error*

Order

- The higher the order, the quicker an algorithm

- Secant and Bisection might converge faster, but both can also diverge. It’s the reason why bisection is still used.

[1] Bisection

[2] Secant

[3] Bisection

2.5 Accelerating Convergence 86

2.6 Zeros of Polynomials and Muller's Method 91

CALCULATOR: <https://atozmath.com/CONM/Bisection.aspx?q=mu&q1=x%5e4-2x%5e3-5x%5e2%2b12x-5%60%60false%603%2c1%2c2%604%601%603&dp=4&do=1#tblSolution>

- One main distinguishing factor for Muller’s method is that it requires three initial values , and for the iterations

**Lesson 0**

C10: Newton’s Method for Nonlinear Systems of Equations

[Q] Give two examples of methods which find an approximate solution to by using a non-linear (2) approximation of the function .

We have the Muller’s method, the fixed-point method on , the fixed-point method on , also known as the Newton’s method.

[Q] What is the essential difference between direct methods and iterative methods for solving systems.

The essential difference is that **direct methods** are more suitable for small scale problems and lead to the exact solution

whereas **iterative methods** are more suitable for large scale problems and lead to an approximation of the solution.

10.1 Fixed Points for Functions of Several Variables 642

TODO

10.2 Newton's Method 651

**Newton’s method (for system of non-linear equations)**

1. Define

2. Find Jacobian of

3. Find Inverse

4. Valuation step: Do *recurrence relation*

Example: JUNE2014 Q1 d

Text, letter

Description automatically generated

1. Define

*Make all vectors equal zero*

2. Find Jacobian of

**Wolfram Alpha**

jacobian of (x-x^2-y, xy+2y-1)

3. Find Inverse

**Wolfram Alpha**

inverse of ({1-2x,-1},{y,x+2})

**Inverse of 2x2 matrix**

where

Determinant

4. Valuation step: Do

*recurrence relation*

10.3 Quasi-Newton Methods 659

10.4 Steepest Descent Techniques 666

10.5 Homotopy and Continuation Methods 674 10.6 Numerical Software 682

**Lesson 0**

C6: Direct Methods for Solving Linear Systems

[Q] Determine whether the coefficient matrix is diagonally dominant and deduce whether iterative methods (5) are likely to converge to the solution.

A strictly diagonally dominant matrix A is nonsingular.

*Non-singular iff determinant of the matrix is not zero*

An n × n matrix A is called non-singular or invertible if there exists an matrix B such that

If A does not have an inverse, A is called singular*.*

*Has zero as it’s eigenvalue*

- If the matrix is also diagonally dominant, it will have a unique solution as a linear system.

* Many iterative method require that the matrix A be nonsingular

*e.g. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form Ax = b to obtain its unique solution without row or column interchanges, and the computations will be stable with respect to the growth of round-off errors.*

6.1 Linear Systems of Equations 362

Gaussian Elimination

**1 Gaussian Elimination without pivoting**

*(Naïve Gaussian Elimination algorithm)*

Forward Elimination

* Number of steps of forward elimination is , where is the number of equations

Back Substitution

example: ASS2 Q1

Given

Goal: elements under main diagonal to 0

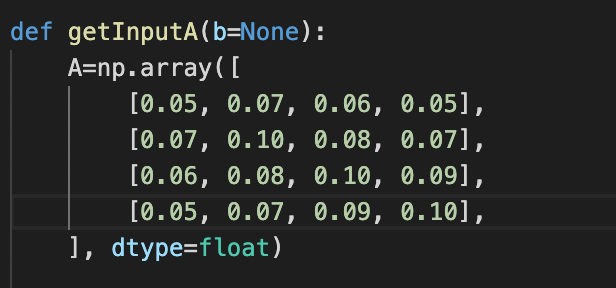
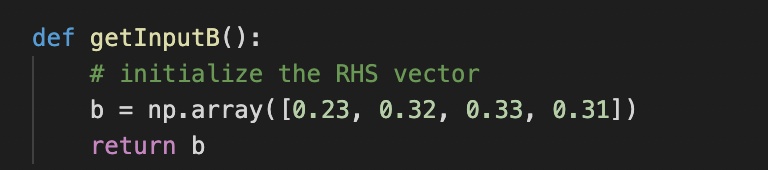
Step 1: Write equation in the form

Augmented matrix

Number of steps:

Scatter chart

Description automatically generated



Matrices\_GaussianEliminationWIthoutPivoting.py

FORWARD ELIMINATION

Step 1: calculate multipliers for each row

*Use 0.05 as the denominator*

*FROMT THIS POINT ON, INCREASE THE PRECISION TO FOUR DECIMAL PLACES AND USE THAT FOR EVERY CALCULATION*

Step 2: Apply algorithm from second row

Step 3: Apply algorithm to third row

*The first column will be 0.00, don’t bother calculating this*

Step 4: Apply algorithm from fourth row

BACK SUBSTITUTION

6.2 Pivoting Strategies

**2 Gaussian Elimination with pivoting**

Pivoting is basically swapping rows with others in the matrix beforehand. This can have a ‘profound’ effect on the results.

Again, the steps to achieve this:

Forward Elimination

* Number of steps of forward elimination is , where is the number of equations

Back Substitution

Compute for each iteration

Compute Compute

**3 Gaussian Elimination with scaled partial pivoting**

Given

:

:

:

Iteration 1:

*the first column*

*computed column that contains the biggest values from each row*

Since the row in with the largest value is , no row interchange is required

*If was the biggest, we would swap with*

:

Iteration 2:

Since the row in with the largest value is , no row interchange is required

:

Iteration 3:

Since the row in with the largest value is , no row interchange is required

6.3 Linear Algebra and Matrix Inversion

TODO

JUNE2015 Q2.1

Schematic

Description automatically generated with medium confidence

**Wolframplha**

({2,-3,1},{1,5/2,-7/2},{-2,-1,-1})

6.4 The Determinant of a Matrix

**4 LU decomposition**

lower–upper (LU) decomposition or factorization

L a lower triangular matrix and

U an upper triangular matrix

ASS2 2014

Step 1:

Step 2:

Step 3:

Step 4:

ASS2 2021

Step 1:

Step 2:

Step 3:

Step 4:

**5 LDU decomposition**

L a lower triangular matrix and

D is a diagonal matrix

U an upper triangular matrix

A Lower-diagonal-upper (LDU) decomposition is a decomposition of the form

6.5 Matrix Factorization 406

6.6 Special Types of Matrices 416

**Lesson 0**

C7: Iterative Techniques in Matrix Algebra

7.1 Norms of Vectors and Matrices

TODO

7.2 Eigenvalues and Eigenvectors

TODO

7.3 The Jacobi and Gauss-Siedel Iterative Techniques

Jacobi Method

<https://www.maa.org/press/periodicals/loci/joma/iterative-methods-for-solving-iaxi-ibi-jacobis-method>

Perhaps the simplest iterative method for solving Ax = b is Jacobi’s Method. **Good**: relatively easy to understand and thus is a good first taste of iterative methods;

**Bad**: because it is not typically used in practice

Still, it is a good starting point for learning about more useful, but more complicated, iterative methods

To choose and , we can use the Jacobi method.

Iterative algorithm.

Each diagonal element is solved for, and an approximate value is plugged in.

The process is then iterated until it converges.

**INPUT:**

initial guess to the solution,

(diagonal dominant) matrix

right-hand side vector

Convergence criterion

**OUTPUT:**

Solution when convergence is reached

The algorithm is formed from a LDU decomposition of a matrix

**Text, letter

Description automatically generated**

Method 2.1: Weighted Jacobi Method

<https://en.wikipedia.org/wiki/Jacobi_method#Python_example>

The weighted Jacobi iteration uses a parameter to compute the iteration as

which means our matrix A is not diagonally dominant.

Therefore, we don't know if the Jacobi method will converge or not.

We have , therefore

initial guess to the solution,`

(diagonal dominant) matrix

right-hand side vector

**example: ASS2 2021 Q2.1 b**

Iteration 1:

Iteration 2:

Iteration 3:

Solution:

[-49.0266 -34.1144 -34.4728 -31.9344]

Error:

[ -8.734426 -12.156534 -12.322124 -11.44533 ]

Method 3: Gauss-Seidel Method

<https://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method>

An improvement on the Jacobi method.

convergence is only guaranteed if the matrix is either strictly diagonally dominant, or symmetric and positive definite

**INPUT:**

(diagonal dominant) matrix

right-hand side vector

**OUTPUT:**

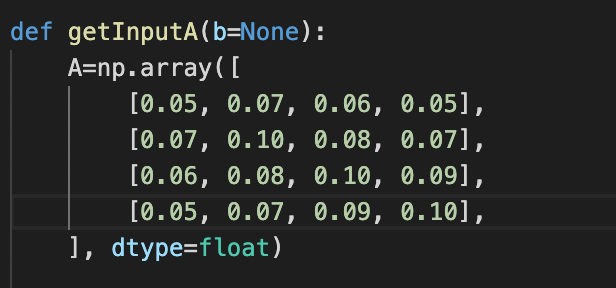
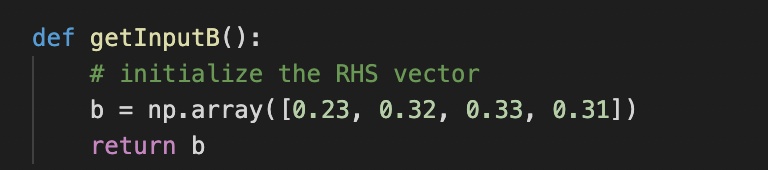
**Text, letter

Description automatically generated**

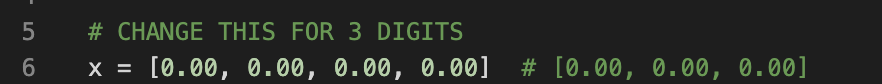
**Gauss-Siedel Method**

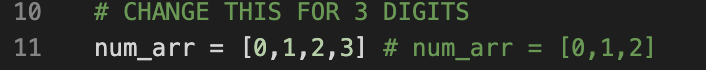
Scatter chart

Description automatically generated

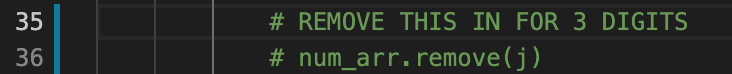


*Also, change the following as 3 or 4 digits depending on if you have a 3x3 or 4x4 matrix*

**

****

**REMOVE!!!**

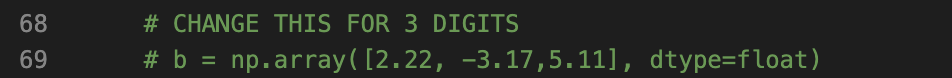
****

**A picture containing text, clock

Description automatically generated**

**Text

Description automatically generated**

****

Matrices\_GaussSiedel.py

**ASS 2 2021 Q2.2 b**

iteration 1:

iteration 2:

iteration 3:

7.4 Relaxation Techniques for Solving Linear Systems

Method 4: Successive Over-Relaxation (SOR)

Variant of Gauss-Seidel, results in faster convergence

<https://en.wikipedia.org/wiki/Successive_over-relaxation>

**INPUT:**

initial guess to the solution,

(diagonal dominant) matrix

right-hand side vector

relaxation factor

**OUTPUT:**

**Text, letter

Description automatically generated**

**Example:**

iteration 1 :

iteration 2 :

iteration 3 :

Method 4.1: Symmetric Successive Over-Relaxation (SSOR)

7.5 Error Bounds and Iterative Refinement

7.6 The Conjugate Gradient Method 487

7.7 Numerical Software 503

**Lesson 0**

C3: Interpolation and polynomial approximation

3.1 Interpolation and the Lagrange Polynomial 104

TODO

3.2 Data Approximation and Neville's Method 115

3.3 Divided Differences (Newton Interpolating polynomial)

DONE: This is the excel table that can be used to calculate this

<https://docs.google.com/spreadsheets/d/1WCpvTcHzQHaiUnwkH7yhUgeWtbfRS5mI/edit?usp=sharing&ouid=104899109488434403761&rtpof=true&sd=true>

JUNE2015 Q3.1

Table

Description automatically generated

Graphical user interface, application, timeline

Description automatically generated

3.4 Hermite Interpolation

Hermite’s Polynomials

(Use Lagrange polynomials and their derivatives)

Hermite’s polynomials using of divided-differences

DONE: This is the excel table that can be used to calculate this

<https://docs.google.com/spreadsheets/d/1WCpvTcHzQHaiUnwkH7yhUgeWtbfRS5mI/edit?usp=sharing&ouid=104899109488434403761&rtpof=true&sd=true>

Table, Excel

Description automatically generated

Example: NOV2014 Q3.1

Table

Description automatically generated

3.5 Cubic Spline Interpolation 142

- Can sometimes be better than Lagrange polynomials

- Unlike Lagrange Polynomials, each point in the line does not change the entire line

- these are function approximations that are continuous at merging points

- also have continuous first and second derivatives where they join

Chart

Description automatically generated with medium confidence

Natural cubic spline: has second derivative equal to zero at end points

How to calculate a cubic spline

Given

|  |  |
| --- | --- |
| Function | Explanation |
|  |  |
|  |  |
|  | Derivatives (should be equal) |
|  | Second Derivatives (should be equal) |
|  | Function at end point should equal zero |
|  | Function at end point should equal zero |

Text, letter

Description automatically generated

Example: <https://www.youtube.com/watch?v=f4iNbNRKZKU>

Given:

Find and that make this a cubic spline

1. Find derivatives

**wolframalpha**

2x^3-3x^2+3x-4

**wolframalpha**

6x^2-6x+3

2. Evaluate functions

*Use piecewise boundries given*

S1:Using 1

S2:Using 1

3. Write out equations

*All the functions and their counterparts should be equal*

TODO

JUNE2015 Q4.1

TODO

Normal equations to find the least-square best fit to approximate the data

JAN2021 Q3c

3.6 Parametric Curves 162

TODO

JUNE2015 4.2

Bezier curves?

3.7 Numerical Software and Chapter Review 168

**Lesson 0**

C4: Numerical Differentiation and Integration

4.1 Numerical Differentiation 172

4.2 Richardson's Extrapolation 183

4.3 Elements of Numerical Integration

Graphical user interface, text, application

Description automatically generated

Errors:

Text

Description automatically generated

Table

Description automatically generated with low confidence

TODO

JUNE2015 Q5.1

Trapezoidal rule

- Integration method

- Approximates the value of a definite integral (area under a curve)

Example: NOV2014 5.1a

COMPOSITE TRAPEZOIDAL RULE

Text, letter

Description automatically generated

0. Write

1. let

2. Get missing terms

*The interal [a,b] should be filled with evenly spaced terms inbetween*

*Multiply these terms by 2 (composite trapezoidal) or 3 (composite simpsons )*

0

1

3. Calculate

is just a modification of the trapezoidal rule,

TODO

JUNE2015 Q5.1

Simpson’s rule

*The error term in Simpson's rule involves the fourth derivative of /, so it gives exact results when applied to any polynomial of degree three or less.*

4.4 Composite Numerical Integration 202

4.5 Romberg Integration 211

4.6 Adaptive Quadrature Methods 219

4.7 Gaussian Quadrature

In Gaussian quadrature, the points for evaluation are chosen in an optimal rather than an equally spaced way.

The nodes in the interval [a,b] and coefficients are chosen to minimize the expected error obtained in the approximation.

To measure this accuracy, we assume that the best choice of these values produces the exact result for the largest class of polynomials, that is, the choice that gives the greatest degree of precision.

arbitrary

lie within [a,b]

Therefor, parameters exist

TODO

JUNE2015 Q5.2

4.8 Multiple Integrals 235

4.9 Improper Integrals 250

4.10 Numerical Software and Chapter Review 256

**Lesson 0**

C8: Discrete Least Squares Approximation

8.1 Discrete Least Squares Approximation 506

DONE: This is the excel table that can be used to calculate this

<https://docs.google.com/spreadsheets/d/1WCpvTcHzQHaiUnwkH7yhUgeWtbfRS5mI/edit?usp=sharing&ouid=104899109488434403761&rtpof=true&sd=true>

Example: NOV 2021

Table

Description automatically generated

Formula (if you have time):

Text, letter

Description automatically generated

USE EXCEL SHEET

Calendar

Description automatically generated

Table headings

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

8.2 Orthogonal Polynomials and Least Squares Approximation 517

8.3 Chebyshev Polynomials and Economization of Power Series 526

8.4 Rational Function Approximation 535

8.5 Trigonometric Polynomial Approximation 545

8.6 Fast Fourier Transforms 555

8.7 Numerical Software 567