# Numerical Methods I

## System of Linear Algebraic Equations: Gauss Elimination with Pivoting

**program** mainGaussEliminationWithPivoting

**implicit** **none**

**integer** :: n

**real**, **dimension**(:,:), **allocatable** :: a

**real**, **dimension**(:), **allocatable** :: x

**integer** :: rowCount

n = 5

**allocate**(a(n, n+1))

**allocate**(x(n))

a(1,1) = 0

a(1,2) = -1

a(1,3) = 2

a(1,4) = -3

a(1,5) = 4

a(1,6) = -38.5

a(2,1) = 2

a(2,2) = 3

a(2,3) = -1

a(2,4) = 5

a(2,5) = -2

a(2,6) = 32.4

a(3,1) = -1

a(3,2) = 3

a(3,3) = 2

a(3,4) = -5

a(3,5) = 1

a(3,6) = -17.9

a(4,1) = 1

a(4,2) = 2

a(4,3) = 1

a(4,4) = 2

a(4,5) = 3

a(4,6) = -13.9

a(5,1) = -4

a(5,2) = -6

a(5,3) = -2

a(5,4) = 8

a(5,5) = -3

a(5,6) = 4.9

**call** gaussEliminationWithPivoting(a, n, x)

**write**(\*,\*) "Solution:"

**do** rowCount = 1, n

**write**(\*,10) x(rowCount)

**end** **do**

10 **format**(f5.2)

**end** **program** mainGaussEliminationWithPivoting

**subroutine** gaussEliminationWithPivoting(aIn, n, x)

**implicit** **none**

**integer**, **external** :: getRowNumberWithMaxValueInColumn

**integer**, **intent**(in) :: n

**real**, **dimension**(n, n+1), **intent**(in) :: aIn

**real**, **dimension**(n), **intent**(out) :: x

**real**, **dimension**(n, n+1) :: a

**integer** :: stepCount, rowCount, columnCount

**integer** :: pivotRow, pivotColumn

**integer** :: rowWithMaxValueInPivotColumn

**real** :: pivot

**real** :: factor

**write**(\*,\*)

**write**(\*,\*) "System of Linear Algebraic Equations"

**write**(\*,\*) "Method: Gauss Elimination With Pivoting"

**write**(\*,\*)

a = aIn

**write**(\*,\*) "The Augmented Matrix"

**call** printMatrix2D(a, n, (n + 1))

**write**(\*,\*) "Part 1: Forward Elimination"

**write**(\*,\*) "(Reduce the coefficient matrix to upper triangular form)"

**write**(\*,\*)

**do** stepCount = 1, (n - 1)

**write**(\*,20) "Step #", stepCount

**write**(\*,30) "Eliminate sub-diagonal elements of column #", stepCount

pivotRow = stepCount

pivotColumn = stepCount

**write**(\*,\*) "Checking if pivoting is necessary..."

**write**(\*,\*) "(Swapping rows such that the pivot is the largest element in pivotColumn)"

rowWithMaxValueInPivotColumn = getRowNumberWithMaxValueInColumn(a, n, (n + 1), pivotColumn, pivotRow)

**if**(*abs*(a(pivotRow, pivotColumn)) < *abs*(a(rowWithMaxValueInPivotColumn, pivotColumn))) **then**

**write**(\*,35) "Pivoting: swapping row ", pivotRow, " with ", rowWithMaxValueInPivotColumn

**call** swapRows(a, n, (n + 1), pivotRow, rowWithMaxValueInPivotColumn)

**else**

**write**(\*,\*) "The pivot is the largest element in the pivot column. Pivoting is not needed."

**end** **if**

pivot = a(pivotRow, pivotColumn)

**do** rowCount = (pivotRow + 1), n

factor = a(rowCount, pivotColumn) / pivot

**do** columnCount = pivotColumn, (n + 1)

a(rowCount, columnCount) = a(rowCount, columnCount) - factor \* a(pivotRow, columnCount)

**end** **do**

**end** **do**

**call** printMatrix2D(a, n, (n + 1))

**end** **do**

**write**(\*,\*) "Part 2: Back Substitution"

**write**(\*,50) "Evaluate x(", n, ")"

x(n) = a(n, n+1) / a(n, n)

**write**(\*,40) "x(n) = ", x(n)

**write**(\*,\*)

**do** rowCount = (n - 1), 1, -1

**write**(\*,50) "Evaluate x(",rowCount, ")"

factor = 0

**do** stepCount = (rowCount + 1), n

factor = factor + a(rowCount, stepCount) \* x(stepCount)

**end** **do**

x(rowCount) = (1 / a(rowCount, rowCount)) \* (a(rowCount, n+1) - factor)

**write**(\*,60) "x(",rowCount, ") = ", x(rowCount)

**write**(\*,\*)

**end** **do**

20 **format**(a6, i1)

30 **format**(a43, i1)

35 **format** (a23, i1, a6, i1)

40 **format**(a6, f5.2)

50 **format**(a11, i1, a1)

60 **format**(a3, i1, a4, f5.2)

**end** **subroutine** gaussEliminationWithPivoting

**integer** **function** getRowNumberWithMaxValueInColumn(matrix, rows, columns, column, rowStart)

**implicit** **none**

**integer**, **intent**(in) :: rows, columns

**integer**, **intent**(in) :: column, rowStart

**real**, **dimension**(rows, columns), **intent**(in) :: matrix

**integer** :: rowCount

**real** :: maxValueInColumn

maxValueInColumn = *abs*(matrix(rowStart, column))

**do** rowCount = (rowStart + 1), rows

**if**(*abs*(matrix(rowCount, column)) > maxValueInColumn) **then**

maxValueInColumn = matrix(rowCount, column)

getRowNumberWithMaxValueInColumn = rowCount

**end** **if**

**end** **do**

**end** **function** getRowNumberWithMaxValueInColumn

**subroutine** swapRows(matrix, rows, columns, row1, row2)

**implicit** **none**

**integer**, **intent**(in) :: rows, columns

**integer**, **intent**(in) :: row1, row2

**real**, **dimension**(rows, columns), **intent**(inout) :: matrix

**integer** :: columnCount

**real** :: swapper

**do** columnCount = 1, columns

swapper = matrix(row1, columnCount)

matrix(row1, columnCount) = matrix(row2, columnCount)

matrix(row2, columnCount) = swapper

**end** **do**

**end** **subroutine** swapRows

**subroutine** printMatrix2D(matrix, rows, columns)

**implicit** **none**

**integer**, **intent**(in) :: rows, columns

**real**, **dimension**(rows, columns), **intent**(in) :: matrix

**integer** :: rowCount, columnCount

**do** rowCount = 1, rows

**do** columnCount = 1, columns

**write**(\*,10, advance='no') matrix(rowCount, columnCount)

**end** **do**

**write**(\*,\*)

**end** **do**

**write**(\*,\*)

10 **format**(f7.2)

**end** **subroutine** printMatrix2D

### Output

System of Linear Algebraic Equations

Method: Gauss Elimination With Pivoting

The Augmented Matrix

1.00 -1.00 2.00 -3.00 4.00 -35.40

2.00 3.00 -1.00 5.00 -2.00 32.40

-1.00 3.00 2.00 -5.00 1.00 -17.90

1.00 2.00 1.00 2.00 3.00 -13.90

-4.00 -6.00 -2.00 8.00 -3.00 4.90

Part 1: Forward Elimination

(Reduce the coefficient matrix to upper triangular form)

Step #1

Eliminate sub-diagonal elements of column #1

Checking if pivoting is necessary...

(Swapping rows such that the pivot is the largest element in pivotColumn)

Pivoting: swapping row 1 with 5

-4.00 -6.00 -2.00 8.00 -3.00 4.90

0.00 0.00 -2.00 9.00 -3.50 34.85

0.00 4.50 2.50 -7.00 1.75 -19.12

0.00 0.50 0.50 4.00 2.25 -12.67

0.00 -2.50 1.50 -1.00 3.25 -34.18

Step #2

Eliminate sub-diagonal elements of column #2

Checking if pivoting is necessary...

(Swapping rows such that the pivot is the largest element in pivotColumn)

Pivoting: swapping row 2 with 3

-4.00 -6.00 -2.00 8.00 -3.00 4.90

0.00 4.50 2.50 -7.00 1.75 -19.12

0.00 0.00 -2.00 9.00 -3.50 34.85

0.00 0.00 0.22 4.78 2.06 -10.55

0.00 0.00 2.89 -4.89 4.22 -44.80

Step #3

Eliminate sub-diagonal elements of column #3

Checking if pivoting is necessary...

(Swapping rows such that the pivot is the largest element in pivotColumn)

Pivoting: swapping row 3 with 5

-4.00 -6.00 -2.00 8.00 -3.00 4.90

0.00 4.50 2.50 -7.00 1.75 -19.12

0.00 0.00 2.89 -4.89 4.22 -44.80

0.00 0.00 0.00 5.15 1.73 -7.10

0.00 0.00 0.00 5.62 -0.58 3.83

Step #4

Eliminate sub-diagonal elements of column #4

Checking if pivoting is necessary...

(Swapping rows such that the pivot is the largest element in pivotColumn)

Pivoting: swapping row 4 with 5

-4.00 -6.00 -2.00 8.00 -3.00 4.90

0.00 4.50 2.50 -7.00 1.75 -19.12

0.00 0.00 2.89 -4.89 4.22 -44.80

0.00 0.00 0.00 5.62 -0.58 3.83

0.00 0.00 0.00 0.00 2.26 -10.62

Part 2: Back Substitution

Evaluate x(5)

x(n) =-4.70

Evaluate x(4)

x(4) = 0.20

Evaluate x(3)

x(3) = -8.30

Evaluate x(2)

x(2) = 2.50

Evaluate x(1)

x(1) = 3.10

Solution by Gauss Elimination with Pivoting:

3.10

2.50

-8.30

0.20

-4.70