The Cholesky Factorization and Pivoting Methods in Gaussian Elimination

Nate DeMaagd, Kurt O'Hearn

MTH 499-02

March 11, 2013

- Cholesky factorization: overview, algorithm, example

Outline

- Cholesky factorization: overview, algorithm, example
- Pivoting methods in Gaussian Elimination: connections to factorizations, types, example

The Cholesky Factorization

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular
- Necessary conditions on A:

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular
- Necessary conditions on A:
  - Real-valued entries ($A \in \mathbb{R}^{n \mathrm{x} n}$)
  - Symmetric ($A = A^T$)
  - Positive definite ($x^T A x > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0_n$)

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular
- Necessary conditions on A:
  - Real-valued entries ($A \in \mathbb{R}^{n\text{x}n}$)
  - Symmetric ($A = A^T$)
  - Positive definite ($x^T Ax > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0_n$)
- Cholesky: special case of LU-factorization

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular
- Necessary conditions on A:
  - Real-valued entries ($A \in \mathbb{R}^{n\text{x}n}$)
  - Symmetric ($A = A^T$)
  - Positive definite ($x^T Ax > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0_n$)
- Cholesky: special case of LU-factorization
- Why are factorizations important? Connections later in pivoting

The Cholesky Factorization

- Given A, we can write A as $A = LL^T$ with L lower triangular
- Necessary conditions on A:
  - Real-valued entries ($A \in \mathbb{R}^{n\mathrm{x}n}$)
  - Symmetric ($A = A^T$)
  - Positive definite ($x^T Ax > 0$ for all $x \in \mathbb{R}^n$, $x \neq 0_n$)
- Cholesky: special case of LU-factorization
- Why are factorizations important? Connections later in pivoting

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow A = LU$ (LU factorization exists)

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1}U^T$

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow$ $A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1}U^T$
- $U(L^T)^{-1}$ upper triangular, $L^{-1}U^T$ lower triangular
  $\Rightarrow$ both sides diagonal (D)

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite $\Rightarrow$ principle minors nonsingular $\Rightarrow$ $A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1} U^T$
- $U(L^T)^{-1}$ upper triangular, $L^{-1}U^T$ lower triangular $\Rightarrow$ both sides diagonal (D)
- $U = DL^T$

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1}U^T$
- $U(L^T)^{-1}$ upper triangular, $L^{-1}U^T$ lower triangular
  $\Rightarrow$ both sides diagonal (D)
- $U = DL^T$
- D has positive diagonal entries $\Rightarrow D = D^{1/2}D^{1/2}$

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow$ $A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1}U^T$
- $U(L^T)^{-1}$ upper triangular, $L^{-1}U^T$ lower triangular
  $\Rightarrow$ both sides diagonal (D)
- $U = DL^T$
- D has positive diagonal entries $\Rightarrow D = D^{1/2}D^{1/2}$
- $A = L^*(L^*)^T$, where $L^* = LD^{1/2}$

Theorem

If A is a real, symmetric, and positive definite matrix, then $A = LL^T$ for a unique upper triangular L.

Proof.

Existence:

- A positive definite $\Rightarrow$ A nonsingular
- Leading principle minors of A positive definite
  $\Rightarrow$ principle minors nonsingular $\Rightarrow A = LU$ (LU factorization exists)
- $LU = A = A^T = U^T L^T \Leftrightarrow U(L^T)^{-1} = L^{-1}U^T$
- $U(L^T)^{-1}$ upper triangular, $L^{-1}U^T$ lower triangular
  $\Rightarrow$ both sides diagonal (D)
- $U = DL^T$
- D has positive diagonal entries $\Rightarrow D = D^{1/2}D^{1/2}$
- $A = L^*(L^*)^T$, where $L^* = LD^{1/2}$

Unicity: straightforward (assume two, show same) $\square$

Cholesky Algorithm

function CHOLESKY$(n, (a_{ij}))$

  for $k = 1$ **to** n **do**

    $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2\right)^{1/2}$

    for $i = k+1$ **to** n **do**

      $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks}\right) / l_{kk}$

  return (l_{ij})

where

- $A = (a_{ij})$: factored matrix
- n: order of A
- $L = (l_{ij})$: resulting lower triangular matrix

Example of Cholesky Algorithm

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} & 0 & 0 \\ & & 0 \\ & & \end{bmatrix}$$

function $\textsc{Cholesky}(n, (a_{ij}))$
    for $k = 1$ **to** n **do**
        $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
        for $i = k+1$ **to** n **do**
            $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
    return (l_{ij})

Example of Cholesky Algorithm

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ & & 0 \\ & & \end{bmatrix}$$

function CHOLESKY$(n, (a_{ij}))$
    for $k = 1$ **to** n **do**
        $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
        for $i = k+1$ **to** n **do**
            $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
    return (l_{ij})

$$k = 1 :$$

$$l_{11} = (25 - 0)^{1/2} = 5$$

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & & 0 \\ & & \end{bmatrix}$$

function CHOLESKY$(n, (a_{ij}))$
    for $k = 1$ **to** n **do**
        $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
        for $i = k+1$ **to** n **do**
            $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
    return (l_{ij})

$$k = 1, \quad i = 2 :$$

$$l_{21} = (15 - 0)/5 = 3$$

Example of Cholesky Algorithm

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & & 0 \\ -1 & & \end{bmatrix}$$

```
function CHOLESKY(n, (a_ij))
    for k = 1 to n do
        l_kk ← ( a_kk − Σ_{s=1}^{k−1} l_ks^2 )^{1/2}
        for i = k + 1 to n do
            l_ik ← ( a_ik − Σ_{s=1}^{k−1} l_is l_ks ) / l_kk
    return (l_ij)
```

$$k = 1, \quad i = 3 :$$

$$l_{31} = (-5 - 0)/5 = -1$$

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & & \end{bmatrix}$$

1: **function** CHOLESKY$(n, (a_{ij}))$
2: **for** $k = 1$ **to** n **do**
3: $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
4: **for** $i = k+1$ **to** n **do**
5: $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
6: **return** (l_{ij})

$$k = 2:$$

$$l_{22} = (18 - 3^2)^{1/2} = 3$$

Example of Cholesky Algorithm

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & \end{bmatrix}$$

1: **function** CHOLESKY$(n, (a_{ij}))$
2: **for** $k = 1$ **to** n **do**
3: $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
4: **for** $i = k + 1$ **to** n **do**
5: $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
6: **return** (l_{ij})

$$k = 2, \quad i = 3:$$

$$l_{32} = (0 - (-1)(3)) / 3 = 1$$

Example of Cholesky Algorithm

$$A = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix}$$

function $\textsc{Cholesky}(n, (a_{ij}))$
    for $k = 1$ **to** n **do**
        $l_{kk} \leftarrow \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2 \right)^{1/2}$
        for $i = k + 1$ **to** n **do**
            $l_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$
    return (l_{ij})

$$k = 3:$$

$$l_{33} = \left(11 - (-1)^2 - (-1)^2\right) = 3$$

- Verify correctness:

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix}$$

Example of Cholesky Algorithm Cont.

- Verify correctness:

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix} \Longrightarrow L^T = \begin{bmatrix} 5 & 3 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

Example of Cholesky Algorithm Cont.

- Verify correctness:

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix} \Longrightarrow L^T = \begin{bmatrix} 5 & 3 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

$$LL^T = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 5 & 3 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = A$$

Gaussian Elimination

Gaussian Elimination

- Process which attempts to solve a linear system by reducing the coefficient matrix to triangular form using elementary row operations (the result can then by easily solved using forward/backward substitution)

Gaussian Elimination

- Process which attempts to solve a linear system by reducing the coefficient matrix to triangular form using elementary row operations (the result can then by easily solved using forward/backward substitution)
- Possible row operations:
  - Interchange
  - Scaling
  - Addition

Gaussian Elimination

- Process which attempts to solve a linear system by reducing the coefficient matrix to triangular form using elementary row operations (the result can then by easily solved using forward/backward substitution)
- Possible row operations:
  - Interchange
  - Scaling
  - Addition
- Gaussian elimination only utilizes scaling and addition!

Gaussian Elimination Algorithm

function GE$(n, (a_{ij}))$
    for $k = 1$ **to** $n - 1$ **do**
        for $i = k + 1$ **to** n **do**
            $z \leftarrow a_{ik}/a_{kk}$
            $a_{ik} \leftarrow 0$
            for $j = k + 1$ **to** n **do**
                $a_{ij} \leftarrow a_{ij} - z a_{kj}$
    return (a_{ij})

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices
  - Example: scale row 2 of $A \in \mathbb{R}^{3x3}$ by 4

$$EA, \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices
  - Example: scale row 2 of $A \in \mathbb{R}^{3x3}$ by 4

$$EA, \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Define G.E. recursively in terms of multiplication by these matrices:

$$A_{k+1} = E_k A_k$$

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices
  - Example: scale row 2 of $A \in \mathbb{R}^{3\text{x}3}$ by 4

$$EA, \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Define G.E. recursively in terms of multiplication by these matrices:

$$A_{k+1} = E_k A_k$$

- Note that algorithm terminates after $s = T_n$ operations with $A_{n\text{x}n}$

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices
  - Example: scale row 2 of $A \in \mathbb{R}^{3x3}$ by 4

$$EA, \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Define G.E. recursively in terms of multiplication by these matrices:

$$A_{k+1} = E_k A_k$$

- Note that algorithm terminates after $s = T_n$ operations with A_{nxn}
- Define $U = A_{s+1}$ and observe that

$$U = E_s \cdots E_1 A \quad \text{or} \quad A = E_1^{-1} \cdots E_s^{-1} U = LU$$

- Recall: row operations are equivalent to left multiplication of coefficient matrix by elementary matrices
  - Example: scale row 2 of $A \in \mathbb{R}^{3\text{x}3}$ by 4

$$EA, \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Define G.E. recursively in terms of multiplication by these matrices:

$$A_{k+1} = E_k A_k$$

- Note that algorithm terminates after $s = T_n$ operations with $A_{n\text{x}n}$
- Define $U = A_{s+1}$ and observe that

$$U = E_s \cdots E_1 A \quad \text{or} \quad A = E_1^{-1} \cdots E_s^{-1} U = LU$$

- So, Gaussian elimination constructs an LU factorization (and hence is functionally equivalent to the LU factorization algorithm!)

Pivoting Methods

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)
- Solution: incorporate row interchange operations via permutation matrix (we call this pivoting)

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)
- Solution: incorporate row interchange operations via permutation matrix (we call this pivoting)
- Types of pivoting:
  - Partial/complete: pertains to the amount of the matrix examined to find where to pivot

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)
- Solution: incorporate row interchange operations via permutation matrix (we call this pivoting)
- Types of pivoting:
  - Partial/complete: pertains to the amount of the matrix examined to find where to pivot
    - Partial: analyze one row
    - Complete: analyze entire submatrix

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)
- Solution: incorporate row interchange operations via permutation matrix (we call this pivoting)
- Types of pivoting:
  - Partial/complete: pertains to the amount of the matrix examined to find where to pivot
    - Partial: analyze one row
    - Complete: analyze entire submatrix
  - Unscaled/scaled: pertains to how the potential pivot locations are compared

Pivoting Methods

- Problem: Gaussian elimination breaks down (spectacular examples pg. 167–169)
- Solution: incorporate row interchange operations via permutation matrix (we call this pivoting)
- Types of pivoting:
  - Partial/complete: pertains to the amount of the matrix examined to find where to pivot
    - Partial: analyze one row
    - Complete: analyze entire submatrix
  - Unscaled/scaled: pertains to how the potential pivot locations are compared
    - Unscaled: compare absolute value of entries directly
    - Scaled: compare ratios of pivot entries to maximum entries in magnitude in each row

```
function GE_PP(n, (a_ij), (p_i))
    for k = 1 to n - 1 do
        for i = k + 1 to n do
            z ← a_{p_i k} / a_{p_k k}
            a_{p_i k} ← 0
            for j = k + 1 to n do
                a_{p_i j} ← a_{p_i j} - z a_{p_k j}
    return (a_ij)
```

Pivoting Examples

$$A_1 = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 1 & 1 \\ 1 & 3 & 3 \end{bmatrix}$$

$$p = \begin{bmatrix} 1, & 2, & 3 \end{bmatrix}$$

```
function GE_PP(n, (a_ij), (p_i))
    for k = 1 to n − 1 do
        for i = k + 1 to n do
            z ← a_{p_i k} / a_{p_k k}
            a_{p_i k} ← 0
            for j = k + 1 to n do
                a_{p_i j} ← a_{p_i j} − z a_{p_k j}
    return (a_ij)
```

$$\max(|1|, |2|, |1|) = 2$$

Pivoting Examples

$$A_1 = \begin{bmatrix} 1 & 3 & 6 \\ 2 & 1 & 1 \\ 1 & 3 & 3 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

function GE_PP($n, (a_{ij}), (p_i)$)
    for $k = 1$ **to** $n - 1$ **do**
        for $i = k + 1$ **to** n **do**
            $z \leftarrow a_{p_i k} / a_{p_k k}$
            $a_{p_i k} \leftarrow 0$
            for $j = k + 1$ **to** n **do**
                $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
    return (a_{ij})

$$\max(|1|, |2|, |1|) = 2$$

$$R_1 \leftarrow R_1 - \frac{1}{2} R_2$$

$$R_3 \leftarrow R_3 - \frac{1}{2} R_2$$

$$R_2 \leftarrow \frac{1}{2} R_2$$

Pivoting Examples

$$A_2 = \begin{bmatrix} 0 & 2.5 & 5.5 \\ 1 & 0.5 & 0.5 \\ 0 & 2.5 & 2.5 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

```
function GE_PP(n, (a_ij), (p_i))
    for k = 1 to n − 1 do
        for i = k + 1 to n do
            z ← a_{p_i k} / a_{p_k k}
            a_{p_i k} ← 0
            for j = k + 1 to n do
                a_{p_i j} ← a_{p_i j} − z a_{p_k j}
    return (a_ij)
```

$$\max(|1|, |2|, |1|) = 2$$

$$R_1 \leftarrow R_1 - \frac{1}{2} R_2$$

$$R_3 \leftarrow R_3 - \frac{1}{2} R_2$$

$$R_2 \leftarrow \frac{1}{2} R_2$$

Pivoting Examples

$$A_2 = \begin{bmatrix} 0 & 2.5 & 5.5 \\ 1 & 0.5 & 0.5 \\ 0 & 2.5 & 2.5 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

```
function GE_PP(n, (a_ij), (p_i))
    for k = 1 to n − 1 do
        for i = k + 1 to n do
            z ← a_{p_i k} / a_{p_k k}
            a_{p_i k} ← 0
            for j = k + 1 to n do
                a_{p_i j} ← a_{p_i j} − z a_{p_k j}
    return (a_ij)
```

$$\max(|2.5|, |2.5|) = 2.5$$

Pivoting Examples

$$A_2 = \begin{bmatrix} 0 & 2.5 & 5.5 \\ 1 & 0.5 & 0.5 \\ 0 & 2.5 & 2.5 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

function GE_PP($n, (a_{ij}), (p_i)$)
  for $k = 1$ **to** $n - 1$ **do**
    for $i = k + 1$ **to** n **do**
      $z \leftarrow a_{p_i k} / a_{p_k k}$
      $a_{p_i k} \leftarrow 0$
      for $j = k + 1$ **to** n **do**
        $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
  return (a_{ij})

$$\max(|2.5|, |2.5|) = 2.5$$

$$R_3 \leftarrow R_3 - R_2$$

$$R_1 \leftarrow \frac{2}{5} R_1$$

Pivoting Examples

$$A_3 = \begin{bmatrix} 0 & 1 & 2.2 \\ 1 & 0.5 & -0.6 \\ 0 & 0 & -3 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

function GE_PP$(n, (a_{ij}), (p_i))$
  for $k = 1$ **to** $n - 1$ **do**
    for $i = k + 1$ **to** n **do**
      $z \leftarrow a_{p_i k} / a_{p_k k}$
      $a_{p_i k} \leftarrow 0$
      for $j = k + 1$ **to** n **do**
        $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
  return (a_{ij})

$$\max(|2.5|, |2.5|) = 2.5$$

$$R_3 \leftarrow R_3 - R_2$$

$$R_1 \leftarrow \frac{2}{5} R_1$$

Pivoting Examples

$$A_3 = \begin{bmatrix} 0 & 1 & 2.2 \\ 1 & 0.5 & \text{-}0.6 \\ 0 & 0 & \text{-}3 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

function GE_PP$(n, (a_{ij}), (p_i))$
  for $k = 1$ **to** $n - 1$ **do**
    for $i = k + 1$ **to** n **do**
      $z \leftarrow a_{p_i k} / a_{p_k k}$
      $a_{p_i k} \leftarrow 0$
      for $j = k + 1$ **to** n **do**
        $a_{p_i j} \leftarrow a_{p_i j} - z a_{p_k j}$
  return (a_{ij})

$$R_3 \leftarrow -\frac{1}{3} R_3$$

Pivoting Examples

$$A_4 = \begin{bmatrix} 0 & 1 & 2.2 \\ 1 & 0.5 & \text{-}0.6 \\ 0 & 0 & 1 \end{bmatrix}$$

$$p = \begin{bmatrix} 2, & 1, & 3 \end{bmatrix}$$

```
function GE_PP(n, (a_ij), (p_i))
    for k = 1 to n − 1 do
        for i = k + 1 to n do
            z ← a_{p_i k} / a_{p_k k}
            a_{p_i k} ← 0
            for j = k + 1 to n do
                a_{p_i j} ← a_{p_i j} − z a_{p_k j}
    return (a_ij)
```

$$R_3 \leftarrow -\frac{1}{3} R_3$$