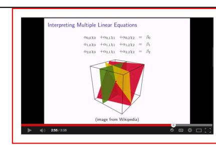
Week 7

More Gaussian Elimination and Matrix Inversion

7.1 Opening Remarks

7.1.1 Introduction



☛ View at edX

7.1.2 Outline

.13

7.1.3 What You Will Learn

Upon completion of this unit, you should be able to

- Determine, recognize, and apply permutation matrices.
- Apply permutation matrices to vectors and matrices.
- Identify and interpret permutation matrices and fluently compute the multiplication of a matrix on the left and right by a permutation matrix.
- Reason, make conjectures, and develop arguments about properties of permutation matrices.
- Recognize when Gaussian elimination breaks down and apply row exchanges to solve the problem when appropriate.
- Recognize when LU factorization fails and apply row pivoting to solve the problem when appropriate.
- Recognize that when executing Gaussian elimination (LU factorization) with $Ax = b$ where A is a square matrix, one of three things can happen:
  1. The process completes with no zeroes on the diagonal of the resulting matrix U. Then $A = LU$ and $Ax = b$ has a unique solution, which can be found by solving $Lz = b$ followed by $Ux = z$.
  2. The process requires row exchanges, completing with no zeroes on the diagonal of the resulting matrix U. Then $PA = LU$ and $Ax = b$ has a unique solution, which can be found by solving $Lz = Pb$ followed by $Ux = z$.
  3. The process requires row exchanges, but at some point no row can be found that puts a nonzero on the diagonal, at which point the process fails (unless the zero appears as the last element on the diagonal, in which case it completes, but leaves a zero on the diagonal of the upper triangular matrix). In Week 8 we will see that this means $Ax = b$ does not have a unique solution.
- Reason, make conjectures, and develop arguments about properties of inverses.
- Find the inverse of a simple matrix by understanding how the corresponding linear transformation is related to the matrix-vector multiplication with the matrix.
- Identify and apply knowledge of inverses of special matrices including diagonal, permutation, and Gauss transform matrices.
- Determine whether a given matrix is an inverse of another given matrix.
- Recognize that a 2×2 matrix $A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} \\ \alpha_{1,0} & \alpha_{1,1} \end{pmatrix}$ has an inverse if and only if its determinant is not zero: $\det(A) = \alpha_{0,0}\alpha_{1,1} - \alpha_{0,1}\alpha_{1,0} \neq 0$.
- Compute the inverse of a 2×2 matrix A if that inverse exists.

Algorithm: $[b] := \text{LTRSV\_UNB\_VAR1}(L, b)$

Partition $L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$

  where L_{TL} is 0×0, b_T has 0 rows

while $m(L_{TL}) < m(L)$ **do**

  Repartition

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

  where λ_{11} is 1×1 , β_1 has 1 row

$b_2 := b_2 - \beta_1 l_{21}$

  Continue with

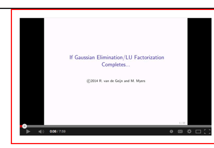
$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Figure 7.1: Algorithm for solving $Lz = b$ when L is a unit lower triangular matrix. The right-hand side vector b is overwritten with the solution vector z.

7.2 When Gaussian Elimination Breaks Down

7.2.1 When Gaussian Elimination Works



☛ **View at edX**

We know that *if* Gaussian elimination completes (the LU factorization of a given matrix can be computed) *and* the upper triangular factor U has no zeroes on the diagonal, then $Ax = b$ can be solved for all right-hand side vectors b.

Why?

- If Gaussian elimination completes (the LU factorization can be computed), then $A = LU$ for some unit lower triangular matrix L and upper triangular matrix U. We know this because of the equivalence of Gaussian elimination and LU factorization.

  If you look at the algorithm for forward substitition (solving $Lz = b$) in Figure 7.1 you notice that the only computations that are encountered are multiplies and adds. Thus, the algorithm will complete.

  Similarly, the backward substitution algorithm (for solving $Ux = z$) in Figure 7.2 can only break down if the division causes an error. And that can only happen if U has a zero on its diagonal.

So, under the mentioned circumstances, we can compute *a* solution to $Ax = b$ via Gaussian elimination, forward substitution, and back substitution. Last week we saw how to compute this solution.

Algorithm: $[b] := \text{UTRSV\_UNB\_VAR1}(U, b)$

Partition $U \rightarrow \left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline U_{BL} & U_{BR} \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$

  where U_{BR} is 0×0, b_B has 0 rows

while $m(U_{BR}) < m(U)$ **do**

 Repartition

$$\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} U_{00} & u_{01} & U_{02} \\ \hline 0 & \upsilon_{11} & u_{12}^T \\ \hline 0 & 0 & U_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

 $\beta_1 := \beta_1 - u_{12}^T b_2$
 $\beta_1 := \beta_1 / \upsilon_{11}$

 Continue with

$$\left(\begin{array}{c|c} U_{TL} & U_{TR} \\ \hline 0 & U_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} U_{00} & u_{01} & U_{02} \\ \hline 0 & \upsilon_{11} & u_{12}^T \\ \hline 0 & 0 & U_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Figure 7.2: Algorithm for solving $Ux = b$ when U is an upper triangular matrix. The right-hand side vector b is overwritten with the solution vector x.

Is this the only solution?

We first give an intuitive explanation, and then we move on and walk you through a rigorous proof.

The reason is as follows: Assume that $Ax = b$ has two solutions: u and v. Then

- $Au = b$ and $Av = b$.
- This then means that vector $w = u - v$ satisfies

$$Aw = A(u - v) = Au - Av = b - b = 0.$$

- Since Gaussian elimination completed we know that

$$(LU)w = 0,$$

  or, equivalently,

$$Lz = 0 \quad \text{and} \quad Uw = z.$$

- It is not hard to see that if $Lz = 0$ then $z = 0$:

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \lambda_{1,0} & 1 & 0 & \cdots & 0 \\ \lambda_{2,0} & \lambda_{2,1} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_{n-1,0} & \lambda_{n-1,1} & \lambda_{n-1,2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} \zeta_0 \\ \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

  means $\zeta_0 = 0$. But then $\lambda_{1,0}\zeta_0 + \zeta_1 = 0$ means $\zeta_1 = 0$. In turn $\lambda_{2,0}\zeta_0 + \lambda_{2,1}\zeta_1 + \zeta_2 = 0$ means $\zeta_2 = 0$. And so forth.

- Thus, $z = 0$ and hence $Uw = 0$.
- It is not hard to see that if $Uw = 0$ then $w = 0$:

$$\begin{pmatrix} \upsilon_{0,0} & \cdots & \upsilon_{0,n-3} & \upsilon_{0,n-2} & \upsilon_{0,n-1} \\ \vdots & \ddots & \vdots & \vdots & \\ 0 & \cdots & \upsilon_{n-3,n-3} & \upsilon_{n-3,n-2} & \upsilon_{n-3,n-1} \\ 0 & \cdots & 0 & \upsilon_{n-2,n-2} & \upsilon_{n-2,n-1} \\ 0 & \cdots & 0 & 0 & \upsilon_{n-11,n-1} \end{pmatrix} \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_{n-3} \\ \omega_{n-2} \\ \omega_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

  means $\upsilon_{n-1,n-1}\omega_{n-1} = 0$ and hence $\omega_{n-1} = 0$ (since $\upsilon_{n-1,n-1} \neq 0$). But then $\upsilon_{n-2,n-2}\omega_{n-2} + \upsilon_{n-2,n-1}\omega_{n-1} = 0$ means $\omega_{n-2} = 0$. And so forth.

We conclude that

> If Gaussian elimination completes with an upper triangular system that has no zero diagonal coefficients (LU factorization computes with L and U where U has no diagonal zero elements), then for all right-hand side vectors, b, the linear system $Ax = b$ has a unique solution x.

A rigorous proof

> Let $A \in \mathbb{R}^{n\times n}$. *If* Gaussian elimination completes and the resulting upper triangular system has no zero coefficients on the diagonal (U has no zeroes on its diagonal), *then* there is a unique solution x to $Ax = b$ for all $b \in \mathbb{R}$.
>
> Always/Sometimes/Never

We don't yet state this as a homework problem, because to get to that point we are going to make a number of observations that lead you to the answer.

Homework 7.2.1.1 Let $L \in \mathbb{R}^{1\times 1}$ be a unit lower triangular matrix. $Lx = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.2 Give the solution of $\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Homework 7.2.1.3 Give the solution of $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.
(Hint: look carefully at the last problem, and you will be able to save yourself some work.)

Homework 7.2.1.4 Let $L \in \mathbb{R}^{2\times 2}$ be a unit lower triangular matrix. $Lx = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.5 Let $L \in \mathbb{R}^{3\times 3}$ be a unit lower triangular matrix. $Lx = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.6 Let $L \in \mathbb{R}^{n\times n}$ be a unit lower triangular matrix. $Lx = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Algorithm: $[b] := \text{LTRSV\_UNB\_VAR2}(L, b)$

Partition $L \rightarrow \left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$

where L_{TL} is 0×0, b_T has 0 rows

while $m(L_{TL}) < m(L)$ **do**

Repartition

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

where λ_{11} is 1×1 , β_1 has 1 row

Continue with

$$\left(\begin{array}{c|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & \lambda_{11} & 0 \\ \hline L_{20} & l_{21} & L_{22} \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Figure 7.3: Blank algorithm for solving $Lx = b$, overwriting b with the result vector x for use in Homework 7.2.1.7. Here L is a lower triangular matrix.

Homework 7.2.1.7 The proof for the last exercise suggests an alternative algorithm (Variant 2) for solving $Lx = b$ when L is unit lower triangular. Use Figure 7.3 to state this alternative algorithm and then implement it, yielding

- `[ b_out ] = Ltrsv_unb_var2( L, b )`

You can check that they compute the right answers with the script in

- `test_Ltrsv_unb_var2.m`

33

Homework 7.2.1.8 Let $L \in \mathbb{R}^{n \times n}$ be a unit lower triangular matrix. $Lx = 0$, where 0 is the zero vector of size n, has the unique solution $x = 0$.

Always/Sometimes/Never

Homework 7.2.1.9 Let $U \in \mathbb{R}^{1 \times 1}$ be an upper triangular matrix with no zeroes on its diagonal. $Ux = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.10 Give the solution of $\begin{pmatrix} -1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Homework 7.2.1.11 Give the solution of $\begin{pmatrix} -2 & 1 & -2 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$.

Homework 7.2.1.12 Let $U \in \mathbb{R}^{2\times 2}$ be an upper triangular matrix with no zeroes on its diagonal. $Ux = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.13 Let $U \in \mathbb{R}^{3\times 3}$ be an upper triangular matrix with no zeroes on its diagonal. $Ux = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

Homework 7.2.1.14 Let $U \in \mathbb{R}^{n\times n}$ be an upper triangular matrix with no zeroes on its diagonal. $Ux = b$, where x is the unknown and b is given, has a unique solution.

Always/Sometimes/Never

The proof for the last exercise closely mirrors how we derived Variant 1 for solving $Ux = b$ last week.

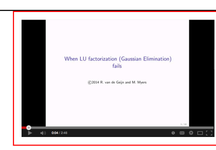
Homework 7.2.1.15 Let $U \in \mathbb{R}^{n\times n}$ be an upper triangular matrix with no zeroes on its diagonal. $Ux = 0$, where 0 is the zero vector of size n, has the unique solution $x = 0$.

Always/Sometimes/Never

Homework 7.2.1.16 Let $A \in \mathbb{R}^{n\times n}$. *If* Gaussian elimination completes and the resulting upper triangular system has no zero coefficients on the diagonal (U has no zeroes on its diagonal), *then* there is a unique solution x to $Ax = b$ for all $b \in \mathbb{R}$.

Always/Sometimes/Never

7.2.2 The Problem



☛ **View at edX**

The question becomes "Does Gaussian elimination always solve a linear system of n equations and n unknowns?" Or, equivalently, can an LU factorization always be computed for an $n \times n$ matrix? In this unit we show that there are linear systems where $Ax = b$ has a unique solution but Gaussian elimination (LU factorization) breaks down. In this and the next sections we will discuss what modifications must be made to Gaussian elimination and LU factorization so that *if* $Ax = b$ has a unique solution, *then* these modified algorithms complete and can be used to solve $Ax = b$.

A simple example where Gaussian elimination and LU factorization break down involves the matrix $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. In the first step, the multiplier equals $1/0$, which will cause a "division by zero" error.

Now, $Ax = b$ is given by the set of linear equations

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \chi_0 \\ \chi_1 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_0 \end{pmatrix}$$

so that $Ax = b$ is equivalent to

$$\begin{pmatrix} \chi_1 \\ \chi_0 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

and the solution to $Ax = b$ is given by the vector $x = \begin{pmatrix} \beta_1 \\ \beta_0 \end{pmatrix}$.

Homework 7.2.2.1 Solve the following linear system, via the steps in Gaussian elimination that you have learned so far.

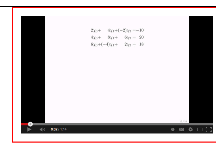
$$\begin{aligned} 2\chi_0 + \quad 4\chi_1 + (-2)\chi_2 &= -10 \\ 4\chi_0 + \quad 8\chi_1 + \quad 6\chi_2 &= 20 \\ 6\chi_0 + (-4)\chi_1 + \quad 2\chi_2 &= 18 \end{aligned}$$

Mark all that are correct:

(a) The process breaks down.

(b) There is no solution.

(c) $\begin{pmatrix} \chi_0 \\ \chi_1 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 4 \end{pmatrix}$



☛ **View at edX**

Now you try an example:

Homework 7.2.2.2 Perform Gaussian elimination with

$$\begin{aligned} 0\chi_0 + \quad 4\chi_1 + (-2)\chi_2 &= -10 \\ 4\chi_0 + \quad 8\chi_1 + \quad 6\chi_2 &= 20 \\ 6\chi_0 + (-4)\chi_1 + \quad 2\chi_2 &= 18 \end{aligned}$$

We now understand how to modify Gaussian elimination so that it completes when a zero is encountered on the diagonal and a nonzero appears somewhere below it.

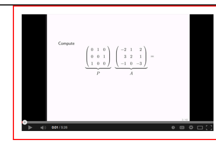
The above examples suggest that the LU factorization algorithm needs to be modified to allow for row exchanges. But to do so, we need to develop some machinery.

7.2.3 Permutations

☛ **View at edX**

Homework 7.2.3.1 Compute

$$\underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}}_{P} \underbrace{\begin{pmatrix} -2 & 1 & 2 \\ 3 & 2 & 1 \\ -1 & 0 & -3 \end{pmatrix}}_{A} =$$



☛ View at edX

Examining the matrix P in the above exercise, we see that each row of P equals a unit basis vector. This leads us to the following definitions that we will use to help express permutations:

Definition 7.1 *A vector with integer components*

$$p = \begin{pmatrix} k_0 \\ k_1 \\ \vdots \\ k_{n-1} \end{pmatrix}$$

is said to be a permutation vector if

- $k_j \in \{0,\ldots,n-1\}$ *, for* $0 \leq j < n$*; and*
- $k_i = k_j$ *implies* $i = j$*.*

In other words, p *is a rearrangement of the numbers* $0,\ldots,n-1$ *(without repetition).*

We will often write $(k_0, k_1, \ldots, k_{n-1})^T$ to indicate the column vector, for space considerations.

Definition 7.2 *Let* $p = (k_0,\ldots,k_{n-1})^T$ *be a permutation vector. Then*

$$P = P(p) = \begin{pmatrix} e_{k_0}^T \\ e_{k_1}^T \\ \vdots \\ e_{k_{n-1}}^T \end{pmatrix}$$

is said to be a permutation matrix.

In other words, P is the identity matrix with its rows rearranged as indicated by the permutation vector $(k_0, k_1, \ldots, k_{n-1})$. We will frequently indicate this permutation matrix as $P(p)$ to indicate that the permutation matrix corresponds to the permutation vector p.

Homework 7.2.3.2 For each of the following, give the permutation matrix $P(p)$:

- If $p = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \end{pmatrix}$ then $P(p) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$,
- If $p = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \end{pmatrix}$ then $P(p) =$
- If $p = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 3 \end{pmatrix}$ then $P(p) =$
- If $p = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 0 \end{pmatrix}$ then $P(p) =$

Homework 7.2.3.3 Let $p = (2,0,1)^T$. Compute

- $P(p) \begin{pmatrix} -2 \\ 3 \\ -1 \end{pmatrix} =$
- $P(p) \begin{pmatrix} -2 & 1 & 2 \\ 3 & 2 & 1 \\ -1 & 0 & -3 \end{pmatrix} =$

Homework 7.2.3.4 Let $p = (2,0,1)^T$ and $P = P(p)$. Compute

$$\begin{pmatrix} -2 & 1 & 2 \\ 3 & 2 & 1 \\ -1 & 0 & -3 \end{pmatrix} P^T =$$

Homework 7.2.3.5 Let $p = (k_0, \ldots, k_{n-1})^T$ be a permutation vector. Consider

$$x = \left(\begin{array}{c} \chi_0 \\ \hline \chi_1 \\ \hline \vdots \\ \hline \chi_{n-1} \end{array} \right).$$

Applying permuation matrix $P = P(p)$ to x yields

$$Px = \left(\begin{array}{c} \chi_{k_0} \\ \hline \chi_{k_1} \\ \hline \vdots \\ \hline \chi_{k_{n-1}} \end{array} \right).$$

Always/Sometimes/Never

Homework 7.2.3.6 Let $p = (k_0, \ldots, k_{n-1})^T$ be a permutation. Consider

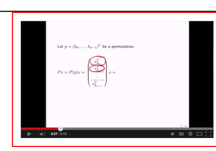
$$A = \left(\begin{array}{c} \widetilde{a}_0^T \\ \hline \widetilde{a}_1^T \\ \hline \vdots \\ \hline \widetilde{a}_{n-1}^T \end{array} \right).$$

Applying $P = P(p)$ to A yields

$$PA = \left(\begin{array}{c} \widetilde{a}_{k_0}^T \\ \hline \widetilde{a}_{k_1}^T \\ \hline \vdots \\ \hline \widetilde{a}_{k_{n-1}}^T \end{array} \right).$$

Always/Sometimes/Never

In other words, Px and PA rearrange the elements of x and the rows of A in the order indicated by permutation vector p.



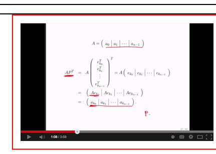
☛ **View at edX**

Homework 7.2.3.7 Let $p = (k_0, \ldots, k_{n-1})^T$ be a permutation, $P = P(p)$, and $A = \left(\begin{array}{c|c|c|c} a_0 & a_1 & \cdots & a_{n-1} \end{array} \right)$.
$AP^T = \left(\begin{array}{c|c|c|c} a_{k_0} & a_{k_1} & \cdots & a_{k_{n-1}} \end{array} \right)$.

Aways/Sometimes/Never

Homework 7.2.3.8 If P is a permutation matrix, then so is P^T.

True/False



☛ **View at edX**

Definition 7.3 *Let us call the special permutation matrix of the form*

$$\widetilde{P}(\pi) = \begin{pmatrix} \boxed{e_\pi^T} \\ e_1^T \\ \vdots \\ e_{\pi-1}^T \\ \boxed{e_0^T} \\ e_{\pi+1}^T \\ \vdots \\ e_{n-1}^T \end{pmatrix} = \left(\begin{array}{c|cccc|c|ccc} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \hline 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ \hline 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{array}\right)$$

a pivot matrix.

$$\widetilde{P}(\pi) = (\widetilde{P}(\pi))^T.$$

Homework 7.2.3.9 Compute

$$\widetilde{P}(1)\begin{pmatrix} -2 \\ 3 \\ -1 \end{pmatrix} = \quad \text{and} \quad \widetilde{P}(1)\begin{pmatrix} -2 & 1 & 2 \\ 3 & 2 & 1 \\ -1 & 0 & -3 \end{pmatrix} = .$$

Homework 7.2.3.10 Compute

$$\begin{pmatrix} -2 & 1 & 2 \\ 3 & 2 & 1 \\ -1 & 0 & -3 \end{pmatrix}\widetilde{P}(1) = .$$

Homework 7.2.3.11 When $\widetilde{P}(\pi)$ (of appropriate size) multiplies a matrix from the left, it swaps row 0 and row π, leaving all other rows unchanged.

Always/Sometimes/Never

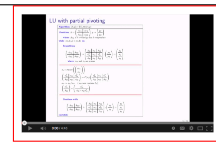
Homework 7.2.3.12 When $\widetilde{P}(\pi)$ (of appropriate size) multiplies a matrix from the right, it swaps column 0 and column π, leaving all other columns unchanged.

Always/Sometimes/Never

7.2.4 Gaussian Elimination with Row Swapping (LU Factorization with Partial Pivoting)

☛ View at edX

Gaussian elimination with row pivoting

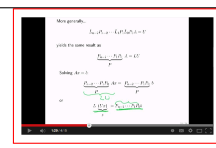


☛ View at edX

We start our discussion with the example in Figure 7.4.

Homework 7.2.4.1 Compute

- $\left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right) \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c|cc} 2 & 4 & -2 \\ \hline 4 & 8 & 6 \\ 6 & -4 & 2 \end{array}\right) =$
- $\left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 3 & 1 & 0 \\ 2 & 0 & 1 \end{array}\right) \left(\begin{array}{c|cc} 2 & 4 & -2 \\ \hline 0 & -16 & 8 \\ 0 & 0 & 10 \end{array}\right) =$
- What do you notice?



☛ View at edX

What the last homework is trying to demonstrate is that, for given matrix A,

- Let $L = \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 3 & 1 & 0 \\ 2 & 0 & 1 \end{array}\right)$ be the matrix in which the multipliers have been collected (the unit lower triangular matrix that has overwritten the strictly lower triangular part of the matrix).
- Let $U = \left(\begin{array}{c|cc} 2 & 4 & -2 \\ \hline 0 & -16 & 8 \\ 0 & 0 & 10 \end{array}\right)$ be the upper triangular matrix that overwrites the matrix.
- Let P be the net result of multiplying all the permutation matrices together, from last to first as one goes from lef t to right:

$$P = \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right) \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right) = \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right)$$

Then

$$PA = LU.$$

In other words, Gaussian elimination with row interchanges computes the LU factorization of a permuted matrix. Of course, one does not generally know ahead of time (*a priori*) what that permutation must be, because one doesn't know when a zero will appear on the diagonal. The key is to notice that when we pivot, we also interchange the multipliers that have overwritten the zeroes that were introduced.

Example 7.4

(You may want to print the blank worksheet at the end of this week so you can follow along.)

In this example, we incorporate the insights from the last two units (Gaussian elimination with row interchanges and permutation matrices) into the explanation of Gaussian elimination that uses Gauss transforms:

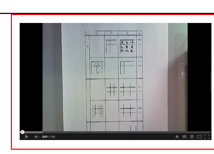
i	L_i	$\tilde{P}$	A	p
0		$\begin{array}{c\|cc} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$	$\begin{array}{c\|cc} 2 & 4 & -2 \\ \hline 4 & 8 & 6 \\ 6 & -4 & 2 \end{array}$	$\begin{array}{c} 0 \\ \hline \cdot \\ \cdot \end{array}$
	$\begin{array}{c\|cc} 1 & 0 & 0 \\ \hline -2 & 1 & 0 \\ -3 & 0 & 1 \end{array}$		$\begin{array}{c\|cc} 2 & 4 & -2 \\ \hline 4 & 8 & 6 \\ 6 & -4 & 2 \end{array}$	$\begin{array}{c} 0 \\ \hline \cdot \\ \cdot \end{array}$
1		$\begin{array}{c\|c\|c} & & \\ \hline & 0 & 1 \\ \hline & 1 & 0 \end{array}$	$\begin{array}{c\|c\|c} 2 & 4 & -2 \\ \hline {\scriptstyle 2} & 0 & 10 \\ \hline {\scriptstyle 3} & -16 & 8 \end{array}$	$\begin{array}{c} 0 \\ \hline 1 \\ \hline \cdot \end{array}$
	$\begin{array}{c\|c\|c} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & -0 & 1 \end{array}$		$\begin{array}{c\|c\|c} 2 & 4 & -2 \\ \hline {\scriptstyle 3} & -16 & 8 \\ \hline {\scriptstyle 2} & 0 & 10 \end{array}$	$\begin{array}{c} 0 \\ \hline 1 \\ \hline \cdot \end{array}$
2			$\begin{array}{cc\|c} 2 & 4 & -2 \\ {\scriptstyle 3} & -16 & 8 \\ \hline {\scriptstyle 2} & {\scriptstyle 0} & 10 \end{array}$	$\begin{array}{c} 0 \\ 1 \\ \hline 0 \end{array}$

Figure 7.4: Example of a linear system that requires row swapping to be added to Gaussian elimination.

Homework 7.2.4.2
(You may want to print the blank worksheet at the end of this week so you can follow along.)
Perform Gaussian elimination with row swapping (row pivoting):

i	L_i	$\tilde{P}$	A	p
0			$\begin{array}{c\|cc} 0 & 4 & -2 \\ \hline 4 & 8 & 6 \\ 6 & -4 & 2 \end{array}$	
1				
2				

The example and exercise motivate the modification to the LU factorization algorithm in Figure 7.5. In that algorithm, PIVOT(x) returns the index of the first nonzero component of x. This means that the algorithm only works if it is always the case that $\alpha_{11} \neq 0$ or vector a_{21} contains a nonzero component.



☛ View at edX

Algorithm: $[A, p] := \text{LU\_PIV}(A, p)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$, $p \rightarrow \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right)$

where A_{TL} is 0×0 and p_T has 0 components

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \rightarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right)$$

$$\pi_1 = \text{PIVOT}\left(\left(\begin{array}{c} \alpha_{11} \\ \hline a_{21} \end{array}\right)\right)$$

$$\left(\begin{array}{c|c|c} a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right) := P(\pi_1) \left(\begin{array}{c|c|c} a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

$a_{21} := a_{21}/\alpha_{11}$ (a_{21} now contains l_{21})

$$\left(\begin{array}{c} a_{12}^T \\ \hline A_{22} \end{array}\right) = \left(\begin{array}{c} a_{12}^T \\ \hline A_{22} - a_{21} a_{12}^T \end{array}\right)$$

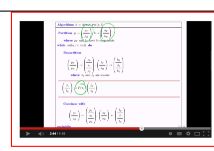
Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \leftarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right)$$

endwhile

Figure 7.5: LU factorization algorithm that incorporates row (partial) pivoting.

Solving the linear system



☛ View at edX

Here is the cool part: We have argued that Gaussian elimination with row exchanges (LU factorization with row pivoting) computes the equivalent of a pivot matrix P and factors L and U (unit lower triangular and upper triangular, respectively) so that $PA = LU$. If we want to solve the system $Ax = b$, then

$$Ax = b$$

is equivalent to

$$PAx = Pb.$$

Now, $PA = LU$ so that

$$\underbrace{(LU)}_{PA} \; x = Pb.$$

Algorithm: $b := \text{APPLY\_PIV}(p, b)$

Partition $p \rightarrow \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$
  where p_T and b_T have 0 components
while $m(b_T) < m(b)$ **do**
  Repartition

$$\left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \rightarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

$$\left(\begin{array}{c} \beta_1 \\ \hline b_2 \end{array}\right) := P(\pi_1) \left(\begin{array}{c} \beta_1 \\ \hline b_2 \end{array}\right)$$

  Continue with

$$\left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \leftarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

Figure 7.6: Algorithm for applying the same exchanges rows that happened during the LU factorization with row pivoting to the components of the right-hand side.

So, solving $Ax = b$ is equivalent to solving

$$L \underbrace{(Ux)}_{z} = Pb.$$

This leaves us with the following steps:

- Update $b := Pb$ by applying the pivot matrices that were encountered during Gaussian elimination with row exchanges to vector b, *in the same order*. A routine that, given the vector with pivot information p, does this is given in Figure 7.6.
- Solve $Lz = b$ with this updated vector b, overwriting b with z. For this we had the routine `Ltrsv_unit`.
- Solve $Ux = b$, overwriting b with x. For this we had the routine `Utrsv_nonunit`.

Uniqueness of solution

If Gaussian elimination with row exchanges (LU factorization with pivoting) completes with an upper triangular system that has no zero diagonal coefficients, then for all right-hand side vectors, b, the linear system $Ax = b$ has a unique solution, x.

7.2.5 When Gaussian Elimination Fails Altogether

Now, we can see that when executing Gaussian elimination (LU factorization) with $Ax = b$ where A is a square matrix, one of three things can happen:

- The process completes with no zeroes on the diagonal of the resulting matrix U. Then $A = LU$ and $Ax = b$ has a unique solution, which can be found by solving $Lz = b$ followed by $Ux = z$.

- The process requires row exchanges, completing with no zeroes on the diagonal of the resulting matrix U. Then $PA = LU$ and $Ax = b$ has a unique solution, which can be found by solving $Lz = Pb$ followed by $Ux = z$.
- The process requires row exchanges, but at some point no row can be found that puts a nonzero on the diagonal, at which point the process fails (unless the zero appears as the last element on the diagonal, in which case it completes, but leaves a zero on the diagonal).

This last case will be studied in great detail in future weeks. For now, we simply state that in this case $Ax = b$ either has *no* solutions, or it has an *infinite* number of solutions.

7.3 The Inverse Matrix

7.3.1 Inverse Functions in 1D

☛ View at edX

In high school, you should have been exposed to the idea of an inverse of a function of one variable. If

- $f : \mathbb{R} \to \mathbb{R}$ maps a real to a real; and
- it is a *bijection* (both one-to-one and onto)

then

- $f(x) = y$ has a unique solution for all $y \in \mathbb{R}$.
- The function that maps y to x so that $g(y) = x$ is called the inverse of f.
- It is denoted by $f^{-1} : \mathbb{R} \to \mathbb{R}$.
- Importantly, $f(f^{-1}(x)) = x$ and $f^{-1}(f(x)) = x$.

In the next units we will examine how this extends to vector functions and linear transformations.

7.3.2 Back to Linear Transformations

☛ View at edX

Theorem 7.5 *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a vector function. Then f is one-to-one and onto (a bijection) implies that $m = n$.*

The proof of this hinges on the dimensionality of $\mathbb{R}^m$ and $\mathbb{R}^n$. We won't give it here.

Corollary 7.6 *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a vector function that is a bijection. Then there exists a function $f^{-1} : \mathbb{R}^n \to \mathbb{R}^n$, which we will call its inverse, such that $f(f^{-1}(x)) = f^{-1}(f(x)) = x$.*

This is an immediate consequence of the fact that for every y there is a unique x such that $f(x) = y$ and $f^{-1}(y)$ can then be defined to equal that x.

Homework 7.3.2.1 Let $L : \mathbb{R}^n \to \mathbb{R}^n$ be a linear transformation that is a bijection and let L^{-1} denote its inverse.

L^{-1} is a linear transformation.

Always/Sometimes/Never

☛ **View at edX**

What we conclude is that if $A \in \mathbb{R}^{n \times n}$ is the matrix that represents a linear transformation that is a bijection L, then there is a matrix, which we will denote by A^{-1}, that represents L^{-1}, the inverse of L. Since for all $x \in \mathbb{R}^n$ it is the case that $L(L^{-1}(x)) = L^{-1}(L(x)) = x$, we know that $AA^{-1} = A^{-1}A = I$, the identity matrix.

Theorem 7.7 *Let $L : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a linear transformation, and let A be the matrix that represents L. If there exists a matrix B such that $AB = BA = I$, then L has an inverse, L^{-1}, and B equals the matrix that represents that linear transformation.*

Actually, it suffices to require there to be a matrix B such that $AB = I$ *or* $BA = I$. But we don't quite have the knowledge at this point to be able to prove it from that weaker assumption.

Proof: We need to show that L is a bijection. Clearly, for every $x \in \mathbb{R}^n$ there is a $y \in \mathbb{R}^n$ such that $y = L(x)$. The question is whether, given any $y \in \mathbb{R}^n$, there is a vector $x \in \mathbb{R}^n$ such that $L(x) = y$. But

$$L(By) = A(By) = (AB)y = Iy = y.$$

So, $x = By$ has the property that $L(x) = y$.

But is this vector x unique? If $Ax_0 = y$ and $Ax_1 = y$ then $A(x_0 - x_1) = 0$. Since $BA = I$ we find that $BA(x_0 - x_1) = x_0 - x_1$ and hence $x_0 - x_1 = 0$, meaning that $x_0 = x_1$.

Let $L : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and let A be the matrix that represents L. Then L has an inverse if and only if there exists a matrix B such that $AB = BA = I$. We will call matrix B the inverse of A, denote it by A^{-1} and note that if $AA^{-1} = I$ then $A^{-1}A = I$.

Definition 7.8 *A matrix A is said to be invertible if the inverse, A^{-1}, exists. An equivalent term for invertible is nonsingular.*

We are going to collect a string of conditions that are equivalent to the statement "A is invertible". Here is the start of that collection.

The following statements are equivalent statements about $A \in \mathbb{R}^{n \times n}$:

- A is nonsingular.
- A is invertible.
- A^{-1} exists.
- $AA^{-1} = A^{-1}A = I$.
- A represents a linear transformation that is a bijection.
- $Ax = b$ has a unique solution for all $b \in \mathbb{R}^n$.
- $Ax = 0$ implies that $x = 0$.

We will add to this collection as the course proceeds.

Homework 7.3.2.2 Let A, B, and C all be $n \times n$ matrices. If $AB = I$ and $CA = I$ then $B = C$.

True/False

7.3.3 Simple Examples

☛ **View at edX**

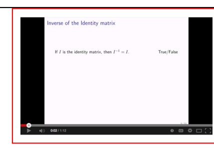
General principles

Given a matrix A for which you want to find the inverse, the first thing you have to check is that A is square. Next, you want to ask yourself the question: "What is the matrix that undoes Ax?" Once you guess what that matrix is, say matrix B, you prove it to yourself by checking that $BA = I$ or $AB = I$.

If that doesn't lead to an answer or if that matrix is too complicated to guess at an inverse, you should use a more systematic approach which we will teach you in the next unit. We will then teach you a fool-proof method next week.

Inverse of the Identity matrix

Homework 7.3.3.1 If I is the identity matrix, then $I^{-1} = I$. True/False



☛ **View at edX**

Inverse of a diagonal matrix

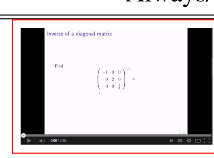
Homework 7.3.3.2 Find

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & \frac{1}{3} \end{pmatrix}^{-1} =$$

Homework 7.3.3.3 Assume $\delta_j \neq 0$ for $0 \leq j < n$.

$$\begin{pmatrix} \delta_0 & 0 & \cdots & 0 \\ 0 & \delta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{n-1} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\delta_0} & 0 & \cdots & 0 \\ 0 & \frac{1}{\delta_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\delta_{n-1}} \end{pmatrix}.$$

Always/Sometimes/Never



☛ **View at edX**

Inverse of a Gauss transform

Homework 7.3.3.4 Find

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}^{-1} =$$

Important: read the answer!

Homework 7.3.3.5

$$\left(\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{21} & I \end{array}\right)^{-1} = \left(\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & -l_{21} & I \end{array}\right).$$

True/False

The observation about how to compute the inverse of a Gauss transform explains the link between Gaussian elimination with Gauss transforms and LU factorization.

Let's review the example from Section 6.2.4:

Before	After
• $\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 & -2 \\ & -10 & 10 \\ & -16 & 8 \end{pmatrix}.$

Before	After
• $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ & -10 & 10 \\ & -16 & 8 \end{pmatrix}$	$\begin{pmatrix} 2 & 4 & -2 \\ & -10 & 10 \\ & & -8 \end{pmatrix}.$

Now, we can summarize the above by

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

Now

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

so that

$$\begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1.6 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

But, given our observations about the inversion of Gauss transforms, this translates to

$$\begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1.6 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

But, miraculously,

$$\begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1.6 & 1 \end{pmatrix}}_{\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1.6 & 1 \end{pmatrix}} \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

But this gives us the LU factorization of the original matrix:

$$\begin{pmatrix} 2 & 4 & -2 \\ 4 & -2 & 6 \\ 6 & -4 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 1.6 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & -2 \\ 0 & -10 & 10 \\ 0 & 0 & -8 \end{pmatrix}.$$

Now, the LU factorization (overwriting the strictly lower triangular part of the matrix with the multipliers) yielded

$$\begin{pmatrix} 2 & 4 & -2 \\ {\scriptstyle 2} & -10 & 10 \\ {\scriptstyle 3} & {\scriptstyle 1.6} & -8 \end{pmatrix}.$$

NOT a coincidence!

The following exercise explains further:

Homework 7.3.3.6 Assume the matrices below are partitioned conformally so that the multiplications and comparison are legal.

$$\left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & 1 & 0 \\ \hline L_{20} & 0 & I \end{array}\right) \left(\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{21} & I \end{array}\right) = \left(\begin{array}{c|c|c} L_{00} & 0 & 0 \\ \hline l_{10}^T & 1 & 0 \\ \hline L_{20} & l_{21} & I \end{array}\right)$$

Always/Sometimes/Never

☛ **View at edX**

Inverse of a permutation

Homework 7.3.3.7 Find
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} =$$

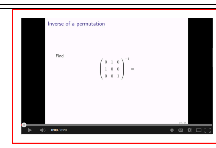
Homework 7.3.3.8 Find
$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{-1} =$$

Homework 7.3.3.9 Let P be a permutation matrix. Then $P^{-1} = P$.

Always/Sometimes/Never

Homework 7.3.3.10 Let P be a permutation matrix. Then $P^{-1} = P^T$.

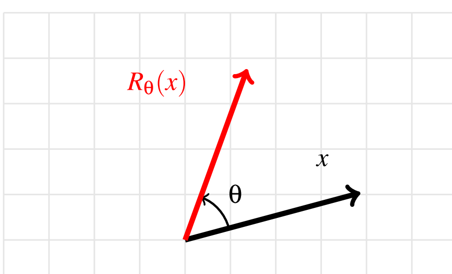
Always/Sometimes/Never



☛ **View at edX**

Inverting a 2D rotation

Homework 7.3.3.11 Recall from Week 2 how $R_\theta(x)$ rotates a vector x through angle θ:



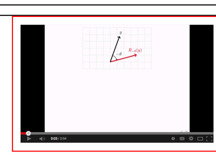
R_θ is represented by the matrix
$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

What transformation will "undo" this rotation through angle θ? (Mark all correct answers)

(a) $R_{-\theta}(x)$

(b) Ax, where $A = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix}$

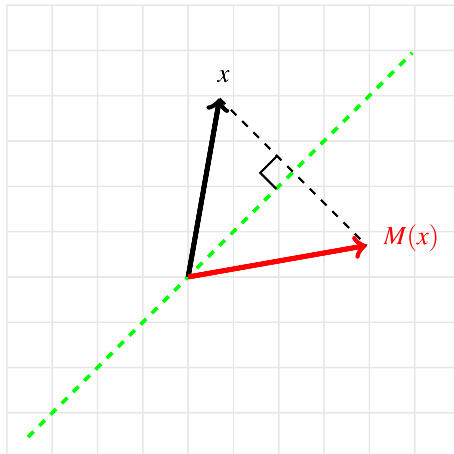
(c) Ax, where $A = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$



☛ **View at edX**

Inverting a 2D reflection

Homework 7.3.3.12 Consider a reflection with respect to the 45 degree line:



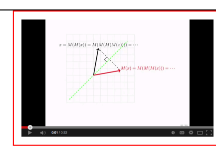
If A represents the linear transformation M, then

(a) $A^{-1} = -A$

(b) $A^{-1} = A$

(c) $A^{-1} = I$

(d) All of the above.



☛ View at edX

7.3.4 More Advanced (but Still Simple) Examples

☛ View at edX

More general principles

Notice that $AA^{-1} = I$. Let's label A^{-1} with the letter B instead. Then $AB = I$. Now, partition both B and I by columns. Then

$$A \left(\begin{array}{c|c|c|c} b_0 & b_1 & \cdots & b_{n-1} \end{array} \right) = \left(\begin{array}{c|c|c|c} e_0 & e_1 & \cdots & e_{n-1} \end{array} \right)$$

and hence $Ab_j = e_j$. So.... the jth column of the inverse equals the solution to $Ax = e_j$ where A and e_j are input, and x is output.

We can now add to our string of equivalent conditions:

The following statements are equivalent statements about $A \in \mathbb{R}^{n \times n}$:

- A is nonsingular.
- A is invertible.
- A^{-1} exists.
- $AA^{-1} = A^{-1}A = I$.
- A represents a linear transformation that is a bijection.
- $Ax = b$ has a unique solution for all $b \in \mathbb{R}^n$.
- $Ax = 0$ implies that $x = 0$.
- $Ax = e_j$ has a solution for all $j \in \{0, \ldots, n-1\}$.

Inverse of a triangular matrix

Homework 7.3.4.1 Compute $\begin{pmatrix} -2 & 0 \\ 4 & 2 \end{pmatrix}^{-1} =$

☛ **View at edX**

Homework 7.3.4.2 Find

$$\begin{pmatrix} 1 & -2 \\ 0 & 2 \end{pmatrix}^{-1} =$$

Homework 7.3.4.3 Let $\alpha_{0,0} \neq 0$ and $\alpha_{1,1} \neq 0$. Then

$$\begin{pmatrix} \alpha_{0,0} & 0 \\ \alpha_{1,0} & \alpha_{1,1} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\alpha_{0,0}} & 0 \\ -\frac{\alpha_{1,0}}{\alpha_{0,0}\alpha_{1,1}} & \frac{1}{\alpha_{1,1}} \end{pmatrix}$$

True/False

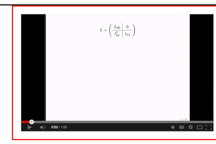
Homework 7.3.4.4 Partition lower triangular matrix L as

$$L = \left(\begin{array}{c|c} L_{00} & 0 \\ \hline l_{10}^T & \lambda_{11} \end{array}\right)$$

Assume that L has no zeroes on its diagonal. Then

$$L^{-1} = \left(\begin{array}{c|c} L_{00}^{-1} & 0 \\ \hline -\frac{1}{\lambda_{11}} l_{10}^T L_{00}^{-1} & \frac{1}{\lambda_{11}} \end{array}\right)$$

True/False



☛ **View at edX**

Homework 7.3.4.5 The inverse of a lower triangular matrix with no zeroes on its diagonal is a lower triangular matrix.

True/False

Challenge 7.3.4.6 The answer to the last exercise suggests an algorithm for inverting a lower triangular matrix. See if you can implement it!

Inverting a 2×2 matrix

Homework 7.3.4.7 Find

$$\begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^{-1} =$$

Homework 7.3.4.8 If $\alpha_{0,0}\alpha_{1,1} - \alpha_{1,0}\alpha_{0,1} \neq 0$ then

$$\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} \\ \alpha_{1,0} & \alpha_{1,1} \end{pmatrix}^{-1} = \frac{1}{\alpha_{0,0}\alpha_{1,1} - \alpha_{1,0}\alpha_{0,1}} \begin{pmatrix} \alpha_{1,1} & -\alpha_{0,1} \\ -\alpha_{1,0} & \alpha_{0,0} \end{pmatrix}$$

(Just check by multiplying... Deriving the formula is time consuming.)

True/False

Homework 7.3.4.9 The 2×2 matrix $A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} \\ \alpha_{1,0} & \alpha_{1,1} \end{pmatrix}$ has an inverse if and only if $\alpha_{0,0}\alpha_{1,1} - \alpha_{1,0}\alpha_{0,1} \neq 0$.

True/False

☛ **View at edX**

The expression $\alpha_{0,0}\alpha_{1,1} - \alpha_{1,0}\alpha_{0,1} \neq 0$ is known as the *determinant* of

$$\begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} \\ \alpha_{1,0} & \alpha_{1,1} \end{pmatrix}.$$

This 2×2 matrix has an inverse if and only if its determinant is nonzero. We will see how the determinant is useful again late in the course, when we discuss how to compute eigenvalues of small matrices. The determinant of a $n \times n$ matrix can be defined and is similarly a condition for checking whether the matrix is invertible. For this reason, we add it to our list of equivalent conditions:

The following statements are equivalent statements about $A \in \mathbb{R}^{n \times n}$:

- A is nonsingular.
- A is invertible.
- A^{-1} exists.
- $AA^{-1} = A^{-1}A = I$.
- A represents a linear transformation that is a bijection.
- $Ax = b$ has a unique solution for all $b \in \mathbb{R}^n$.
- $Ax = 0$ implies that $x = 0$.
- $Ax = e_j$ has a solution for all $j \in \{0, \ldots, n-1\}$.
- The determinant of A is nonzero: $\det(A) \neq 0$.

7.3.5 Properties

Inverse of product

Homework 7.3.5.1 Let $\alpha \neq 0$ and B have an inverse. Then

$$(\alpha B)^{-1} = \frac{1}{\alpha} B^{-1}.$$

True/False

Homework 7.3.5.2 Which of the following is true regardless of matrices A and B (as long as they have an inverse and are of the same size)?

(a) $(AB)^{-1} = A^{-1}B^{-1}$

(b) $(AB)^{-1} = B^{-1}A^{-1}$

(c) $(AB)^{-1} = B^{-1}A$

(d) $(AB)^{-1} = B^{-1}$

Homework 7.3.5.3 Let square matrices $A, B, C \in \mathbb{R}^{n \times n}$ have inverses A^{-1}, B^{-1}, and C^{-1}, respectively. Then $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$.

Always/Sometimes/Never

Inverse of transpose

Homework 7.3.5.4 Let square matrix A have inverse A^{-1}. Then $(A^T)^{-1} = (A^{-1})^T$.

Always/Sometimes/Never

Inverse of inverse

Homework 7.3.5.5

$$(A^{-1})^{-1} = A$$

Always/Sometimes/Never

7.4 Enrichment

7.4.1 Library Routines for LU with Partial Pivoting

Various linear algebra software libraries incorporate LU factorization with partial pivoting.

LINPACK

The first such library was LINPACK:

> J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart.
> LINPACK Users' Guide.
> SIAM, 1979.

A link to the implementation of the routine `DGEFA` can be found at

http://www.netlib.org/linpack/dgefa.f.

You will notice that it is written in Fortran and uses what are now called Level-1 BLAS routines. LINPACK preceded the introduction of computer architectures with cache memories, and therefore no blocked algorithm is included in that library.

LAPACK

LINPACK was replaced by the currently most widely used library, LAPACK:

> E. Anderson, Z. Bai, J. Demmel, J. J. Dongarra, J. Ducroz, A. Greenbaum, S. Hammarling, A. E. McKenney, S. Ostroucho, and D. Sorensen.
> LAPACK Users' Guide.
> SIAM 1992.

> E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. Ducroz, A. Greenbaum, S. Hammarling, A. E. McKenney, S. Ostroucho, and D. Sorensen.
> LAPACK Users' Guide (3rd Edition).
> SIAM 1999.

Implementations in this library include

- DGETF2 (unblocked LU factorization with partial pivoting).
- DGETRF (blocked LU factorization with partial pivoting).

It, too, is written in Fortran. The unblocked implementation makes calls to Level-1 (vector-vector) and Level-2 (matrix-vector) BLAS routines. The blocked implementation makes calls to Level-3 (matrix-matrix) BLAS routines. See if you can recognize some of the names of routines.

ScaLAPACK

ScaLAPACK is version of LAPACK that was (re)written for large distributed memory architectures. The design decision was to make the routines in ScaLAPACK reflect as closely as possible the corresponding routines in LAPACK.

L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley.
ScaLAPACK Users' Guilde.
SIAM, 1997.

Implementations in this library include

- PDGETRF (blocked LU factorization with partial pivoting).

ScaLAPACK is wirtten in a mixture of Fortran and C. The unblocked implementation makes calls to Level-1 (vector-vector) and Level-2 (matrix-vector) BLAS routines. The blocked implementation makes calls to Level-3 (matrix-matrix) BLAS routines. See if you can recognize some of the names of routines.

`libflame`

We have already mentioned libflame. It targets sequential and multithreaded architectures.

F. G. Van Zee, E. Chan, R. A. van de Geijn, E. S. Quintana-Orti, G. Quintana-Orti.
The libflame Library for Dense Matrix Computations.
IEEE Computing in Science and Engineering, Vol. 11, No 6, 2009.

F. G. Van Zee.
libflame: The Complete Reference.
www.lulu.com , 2009
(Available from http://www.cs.utexas.edu/ flame/web/FLAMEPublications.html.)

It uses an API so that the code closely resembles the code that you have been writing.

- Various unblocked and blocked implementations.

Elemental

Elemental is a library that targets distributed memory architectures, like ScaLAPACK does.

Jack Poulson, Bryan Marker, Robert A. van de Geijn, Jeff R. Hammond, Nichols A. Romero. Elemental: A New Framework for Distributed Memory Dense Matrix Computations. ACM Transactions on Mathematical Software (TOMS), 2013.
(Available from http://www.cs.utexas.edu/ flame/web/FLAMEPublications.html.)

It is coded in C++ in a style that resembles the FLAME APIs.

- Blocked implementation.

7.5 Wrap Up

7.5.1 Homework

(No additional homework this week.)

7.5.2 Summary

Permutations

Definition 7.9 *A vector with integer components*

$$p = \begin{pmatrix} k_0 \\ k_1 \\ \vdots \\ k_{n-1} \end{pmatrix}$$

is said to be a permutation vector if

- $k_j \in \{0, \ldots, n-1\}$ *, for* $0 \le j < n$*; and*
- $k_i = k_j$ *implies* $i = j$.

In other words, p *is a rearrangement of the numbers* $0, \ldots, n-1$ *(without repetition).*

Definition 7.10 *Let* $p = (k_0, \ldots, k_{n-1})^T$ *be a permutation vector. Then*

$$P = P(p) = \begin{pmatrix} e_{k_0}^T \\ e_{k_1}^T \\ \vdots \\ e_{k_{n-1}}^T \end{pmatrix}$$

is said to be a permutation matrix.

Theorem 7.11 *Let* $p = (k_0, \ldots, k_{n-1})^T$ *be a permutation vector. Consider*

$$P = P(p) = \begin{pmatrix} e_{k_0}^T \\ e_{k_1}^T \\ \vdots \\ e_{k_{n-1}}^T \end{pmatrix}, \quad x = \begin{pmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_{n-1} \end{pmatrix}, \quad \text{and} \quad A = \begin{pmatrix} a_0^T \\ a_1^T \\ \vdots \\ a_{n-1}^T \end{pmatrix}.$$

Then

$$Px = \begin{pmatrix} \chi_{k_0} \\ \chi_{k_1} \\ \vdots \\ \chi_{k_{n-1}} \end{pmatrix}, \quad \text{and} \quad PA = \begin{pmatrix} a_{k_0}^T \\ a_{k_1}^T \\ \vdots \\ a_{k_{n-1}}^T \end{pmatrix}.$$

Theorem 7.12 *Let* $p = (k_0, \ldots, k_{n-1})^T$ *be a permutation vector. Consider*

$$P = P(p) = \begin{pmatrix} e_{k_0}^T \\ e_{k_1}^T \\ \vdots \\ e_{k_{n-1}}^T \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-1} \end{pmatrix}.$$

Then

$$AP^T = \begin{pmatrix} a_{k_0} & a_{k_1} & \cdots & a_{k_{n-1}} \end{pmatrix}.$$

Theorem 7.13 *If* P *is a permutation matrix, so is* P^T.

Definition 7.14 *Let us call the special permutation matrix of the form*

$$\widetilde{P}(\pi) = \begin{pmatrix} \boxed{e_\pi^T} \\ e_1^T \\ \vdots \\ e_{\pi-1}^T \\ \boxed{e_0^T} \\ e_{\pi+1}^T \\ \vdots \\ e_{n-1}^T \end{pmatrix} = \left(\begin{array}{c|ccc|c|ccc} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \hline 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ \hline 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{array}\right)$$

a pivot matrix.

Theorem 7.15 *When $\widetilde{P}(\pi)$ (of appropriate size) multiplies a matrix from the left, it swaps row 0 and row π, leaving all other rows unchanged.*

When $\widetilde{P}(\pi)$ (of appropriate size) multiplies a matrix from the right, it swaps column 0 and column π, leaving all other columns unchanged.

LU with row pivoting

Algorithm: $[A,p] := \text{LU\_PIV}(A,p)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$, $p \rightarrow \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right)$

where A_{TL} is 0×0 and p_T has 0 components

while $m(A_{TL}) < m(A)$ **do**

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \rightarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right)$$

$$\pi_1 = \text{PIVOT}\left(\left(\begin{array}{c} \alpha_{11} \\ \hline a_{21} \end{array}\right)\right)$$

$$\left(\begin{array}{c|c|c} a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right) := P(\pi_1)\left(\begin{array}{c|c|c} a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right)$$

$a_{21} := a_{21}/\alpha_{11}$ (a_{21} now contains l_{21})

$$\left(\begin{array}{c} a_{12}^T \\ \hline A_{22} \end{array}\right) = \left(\begin{array}{c} a_{12}^T \\ \hline A_{22} - a_{21}a_{12}^T \end{array}\right)$$

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & a_{01} & A_{02} \\ \hline a_{10}^T & \alpha_{11} & a_{12}^T \\ \hline A_{20} & a_{21} & A_{22} \end{array}\right), \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \leftarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right)$$

endwhile

Algorithm: $b := \text{APPLY\_PIV}(p,b)$

Partition $p \rightarrow \left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right)$, $b \rightarrow \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right)$

where p_T and b_T have 0 components

while $m(b_T) < m(b)$ **do**

Repartition

$$\left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \rightarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \rightarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

$$\left(\begin{array}{c} \beta_1 \\ \hline b_2 \end{array}\right) := P(\pi_1)\left(\begin{array}{c} \beta_1 \\ \hline b_2 \end{array}\right)$$

Continue with

$$\left(\begin{array}{c} p_T \\ \hline p_B \end{array}\right) \leftarrow \left(\begin{array}{c} p_0 \\ \hline \pi_1 \\ \hline p_2 \end{array}\right), \left(\begin{array}{c} b_T \\ \hline b_B \end{array}\right) \leftarrow \left(\begin{array}{c} b_0 \\ \hline \beta_1 \\ \hline b_2 \end{array}\right)$$

endwhile

- LU factorization with row pivoting, starting with a square nonsingular matrix A, computes the LU factorization of a permuted matrix A: $PA = LU$ (via the above algorithm LU_PIV).

- $Ax = b$ then can be solved via the following steps:

  - Update $b := Pb$ (via the above algorithm APPLY_PIV).
  - Solve $Lz = b$, overwriting b with z (via the algorithm from 6.3.2).
  - Solve $Ux = b$, overwriting b with x (via the algorithm from 6.3.3).

Theorem 7.16 *Let $L : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a linear transformation that is a bijection. Then the inverse function $L^{-1} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ exists and is a linear transformation.*

Theorem 7.17 *If A has an inverse, A^{-1}, then A^{-1} is unique.*

Inverses of special matrices

<table>
<tr><th>Type</th><th>A</th><th>A^{-1}</th></tr>
<tr><td>Identity matrix</td><td>$I=\left(\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{array}\right)$</td><td>$I=\left(\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{array}\right)$</td></tr>
<tr><td>Diagonal matrix</td><td>$D=\left(\begin{array}{cccc} \delta_{0,0} & 0 & \cdots & 0 \\ 0 & \delta_{1,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{n-1,n-1} \end{array}\right)$</td><td>$D^{-1}=\left(\begin{array}{cccc} \delta_{0,0}^{-1} & 0 & \cdots & 0 \\ 0 & \delta_{1,1}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \delta_{n-1,n-1}^{-1} \end{array}\right)$</td></tr>
<tr><td>Gauss transform</td><td>$\widetilde{L}=\left(\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{21} & I \end{array}\right)$</td><td>$\widetilde{L}^{-1}=\left(\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & -l_{21} & I \end{array}\right).$</td></tr>
<tr><td>Permutation matrix</td><td>P</td><td>P^T</td></tr>
<tr><td>2D Rotation</td><td>$R=\left(\begin{array}{cc} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{array}\right)$</td><td>$R^{-1}=\left(\begin{array}{cc} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{array}\right)=R^T$</td></tr>
<tr><td>2D Reflection</td><td>A</td><td>A</td></tr>
<tr><td>Lower triangular matrix</td><td>$L=\left(\begin{array}{c|c} L_{00} & 0 \\ \hline l_{10}^T & \lambda_{11} \end{array}\right)$</td><td>$L^{-1}=\left(\begin{array}{c|c} L_{00}^{-1} & 0 \\ \hline -\frac{1}{\lambda_{11}} l_{10}^T L_{00}^{-1} & \frac{1}{\lambda_{11}} \end{array}\right)$</td></tr>
<tr><td>Upper triangular matrix</td><td>$U=\left(\begin{array}{c|c} U_{00} & u_{01} \\ \hline 0 & \upsilon_{11} \end{array}\right)$</td><td>$U^{-1}=\left(\begin{array}{c|c} U_{00}^{-1} & -U_{00}^{-1} u_{01}/\upsilon_{11} \\ \hline 0 & \frac{1}{\upsilon_{11}} \end{array}\right)$</td></tr>
<tr><td>General 2×2 matrix</td><td>$\left(\begin{array}{cc} \alpha_{0,0} & \alpha_{0,1} \\ \alpha_{1,0} & \alpha_{1,1} \end{array}\right)$</td><td>$\frac{1}{\alpha_{0,0}\alpha_{1,1}-\alpha_{1,0}\alpha_{0,1}}\left(\begin{array}{cc} \alpha_{1,1} & -\alpha_{0,1} \\ -\alpha_{1,0} & \alpha_{0,0} \end{array}\right)$</td></tr>
</table>

The following matrices have inverses:

- Triangular matrices that have no zeroes on their diagonal.
- Diagonal matrices that have no zeroes on their diagonal.
  (Notice: this is a special class of triangular matrices!).
- Gauss transforms.

  (In Week 8 we will generalize the notion of a Gauss transform to matrices of the form $\left(\begin{array}{c|c|c} I & u_{01} & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & l_{21} & 0 \end{array}\right)$.)
- Permutation matrices.
- 2D Rotations.
- 2D Reflections.

General principle

If $A, B \in \mathbb{R}^{n\times n}$ and $AB = I$, then $Ab_j = e_j$, where b_j is the jth column of B and e_j is the jth unit basis vector.

Properties of the inverse

Assume A, B, and C are square matrices that are nonsingular. Then

- $(\alpha B)^{-1} = \frac{1}{\alpha} B^{-1}$.
- $(AB)^{-1} = B^{-1}A^{-1}$.
- $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$.
- $(A^T)^{-1} = (A^{-1})^T$.
- $(A^{-1})^{-1} = A$.

The following statements are equivalent statements about $A \in \mathbb{R}^{n \times n}$:

- A is nonsingular.
- A is invertible.
- A^{-1} exists.
- $AA^{-1} = A^{-1}A = I$.
- A represents a linear transformation that is a bijection.
- $Ax = b$ has a unique solution for all $b \in \mathbb{R}^n$.
- $Ax = 0$ implies that $x = 0$.
- $Ax = e_j$ has a solution for all $j \in \{0, \ldots, n-1\}$.
- The determinant of A is nonzero: $\det(A) \neq 0$.

Blank worksheet for pivoting exercises

i	L_i	$\tilde{P}$	A	p
0				
	$\begin{array}{c\|cc} 1 & 0 & 0 \\ \hline & 1 & 0 \\ & 0 & 1 \end{array}$			
1				
	$\begin{array}{c\|c\|c} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & & 1 \end{array}$			
2				