Matrix Factorization Geometric Algorithms Lecture 13

CAS CS 132

Introduction

Recap Problem

(LAA 4.9.3) On any given day a student is 5% will be ill tomorrow, and 55% of ill students will remain ill tomorrow.

Write down the stochastic matrix for this situation.

Draw the state diagram for this situation.

- healthy or ill. Of the students healthy today,



$\begin{bmatrix} 0.95 & 0.55 \\ 0.05 & 0.45 \end{bmatrix}$



Objectives

- 1. Motivate matrix factorization in general, and the LU factorization in specific
- 2. Recall elementary row operations and connect them to matrices
- 3. Look at the LU factorization, how to find it, and how to use it

Keywords

elementary matrices LU factorization

Motivation

Much of linear algebra is about extending our intuitions about numbers to matrices.

intuitions about numbers to matrices.

For whole numbers, a factor of n is a number m such that m divides n.

Much of linear algebra is about extending our

intuitions about numbers to matrices.

such that m divides n.

2 is a factor of 10, 7 is a factor of 49,...

Much of linear algebra is about extending our

For whole numbers, a factor of n is a number m

For whole numbers, m is a factor of n if there is a number k such that

n = mk

is a number k such that

n can be "split" into m and k. This is called a factorization of n.

For whole numbers, m is a factor of n if there

n = mk

is a number k such that

- factorization of n.

For whole numbers, m is a factor of n if there

n = mk

n can be "split" into m and k. This is called a

 $1 = 2(5), \quad 49 = 7(7), \dots$

An Aside: Polynomials

- We've also likely seen this with polynomials, e.g.
- This is a polynomial factorization.

$x^3 + 6x^2 + 11x + 6 = (x + 1)(x + 2)(x + 3)$

Matrix Factorization

Matrix Factorization

A factorization of a matrix A is an equation which expresses A as a product of one or more matrices, e.g.,

A = BC

Matrix Factorization

A factorization of a matrix A is an equation matrices, e.g.,

to find their product. Factorization is the harder direction.

which expresses A as a product of one or more

A = BC

So far, we've been given two factors and asked

One nice feature of numbers is that they have a <u>unique</u> factorization into <u>prime factors</u>.

unique factorization into prime factors.

One nice feature of numbers is that they have a

There is no such thing for matrices.

<u>unique</u> factorization into <u>prime factors</u>. There is no such thing for matrices. This is a blessing and a curse:

but they tell us different things.

- One nice feature of numbers is that they have a

 - We have more than one kind of factorization



Writing A as the product of multiple matrices can



» make computing with A faster

Writing A as the product of multiple matrices can

- » make computing with A faster
- » make working with A easier

Writing A as the product of multiple matrices can

- Writing A as the product of multiple matrices can
- » make computing with A faster
- » make working with A easier
- » expose important information about A

» make working with A easier

 \gg expose important information about A

Writing A as the product of multiple matrices can » make computing with A faster LU Decomposition

Question. For an matrix A, solve the equations $Ax_1 = b_1$, $Ax_2 = b_2$... $Ax_{k-1} = b_{k-1}$, $Ax_k = b_k$

In other words: we want to solve a bunch of matrix equations over the same matrix.



Question. For a matrix A, solve (for X) in the equation

where X and B are matrices of appropriate dimension.

AX = B

Question. For a matrix A, solve (for X) in the equation

where X and B are matrices of appropriate dimension.

This is (essentially) the same question.

AX = B

Question. Solve AX = B. If A is *invertible*, then we have a solution: Find A^{-1} and then $X = A^{-1}B$.

Question. Solve AX = B. If A is *invertible*, then we have a solution: Find A^{-1} and then $X = A^{-1}B$. What if A^{-1} is not invertible? Even if it is, can we do it faster?

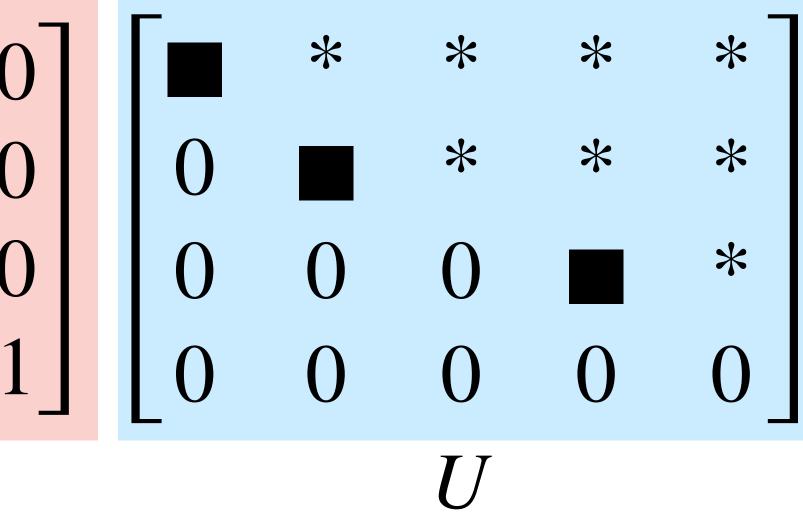
LU Factorization at a High Level

Given a $m \times n$ matrix A, we are going to factorize A as

A

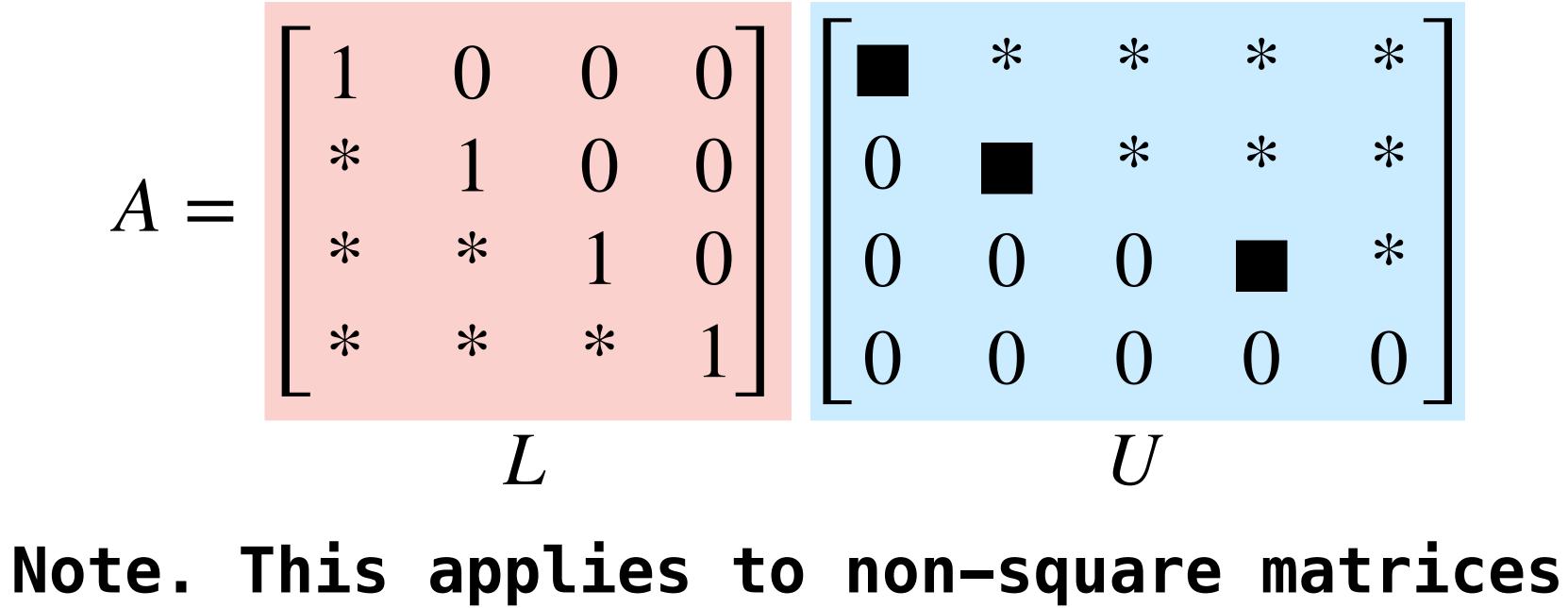
_	[1	0	0	(
	*	1	0	(
	*	*	1	(
	*	*	*	1
		L		

echelon form of A



LU Factorization at a High Level

Given a $m \times n$ matrix A, we are going to factorize A as

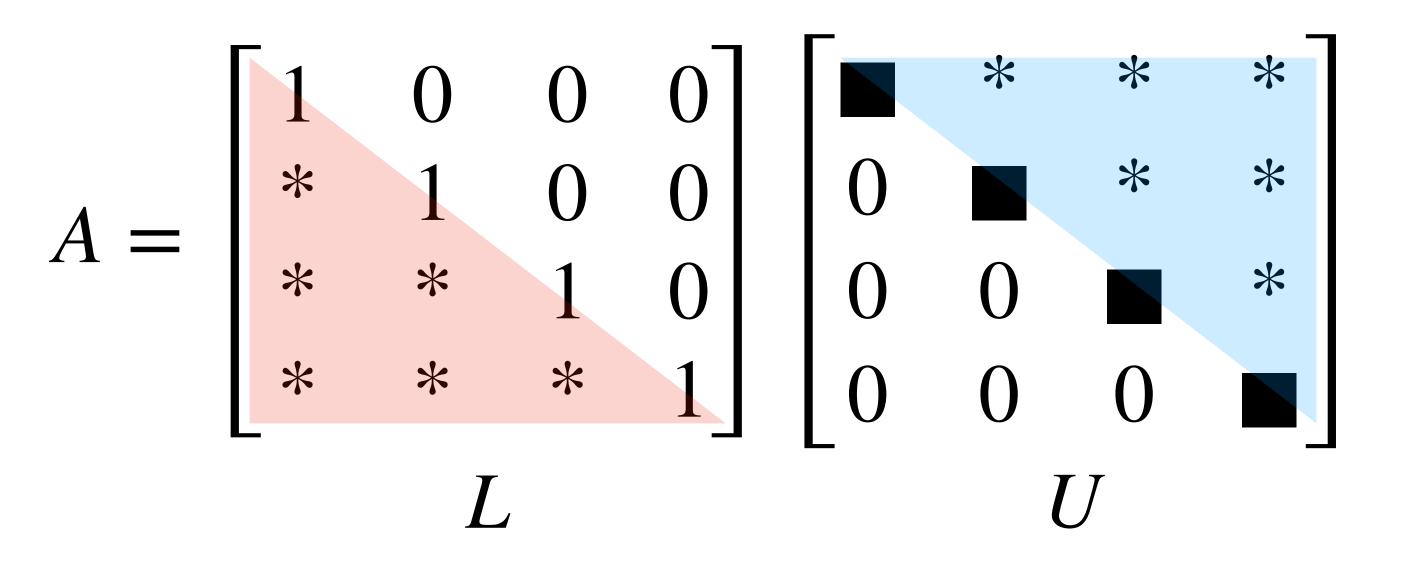


echelon form of A

What are "L" and "U"?

L stands for "lower" as in lower triangular.

U stands for "upper" as in upper triangular. (This only happens when A is square.)



Elementary Matrices



A = UU echelon form of A

We know how to build U, that's just the forward phase of Gaussian elimination.

A = LU echelon form of A

We know how to build *U*, that's just the forward phase of Gaussian elimination. How do we build *L*?

A = LU echelon form of A

We know how to build U_{\bullet} that's just the forward phase of Gaussian elimination.

How do we build L?

The idea. *L* "implements" the row operations of the forward phase.

A = LU echelon form of A

Recall: Elementary Row Operations

- scaling multiply a row by a number interchange switch two rows replacement add two rows (and replace one with the sum) rep. + scl. add a scaled equation to another

The First Key Observation

The First Key Observation

(viewed as transformation on columns)

Elementary row operations are linear transformations

The First Key Observation

(viewed as transformation on columns)

Example: Scale row 2 b

 $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} R_2 \leftarrow$

Elementary row operations are linear transformations

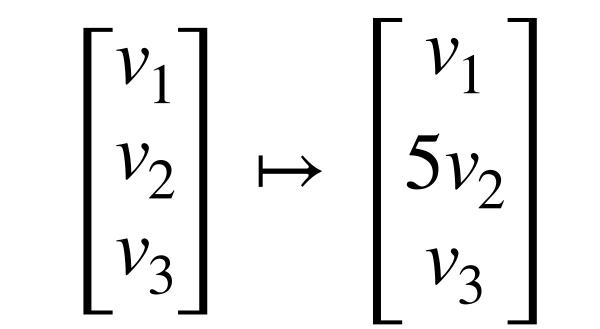
$$\begin{array}{c} \mathbf{y} \ \mathbf{5} \\ -5R_2 \\ \mathbf{-5R}_2 \\ \mathbf{5a_{21}} \ 5a_{22} \ 5a_{23} \\ a_{31} \ a_{32} \ a_{33} \end{array} \right]$$

Example: Scaling

Restricted to one column, we see this is the above transformation.

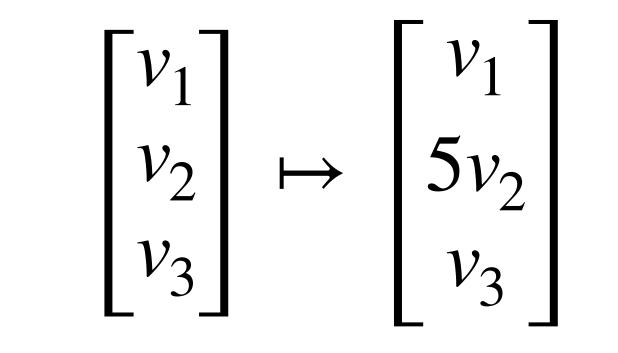
$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \mapsto \begin{bmatrix} v_1 \\ 5v_2 \\ v_3 \end{bmatrix}$

Example: Scaling Let's verify this is linear:



Example: Scaling

Let's build the matrix which implements it:



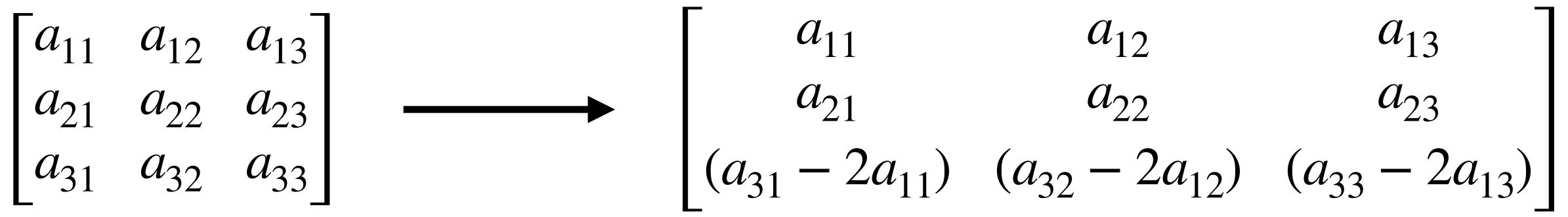
Example: Scaling

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

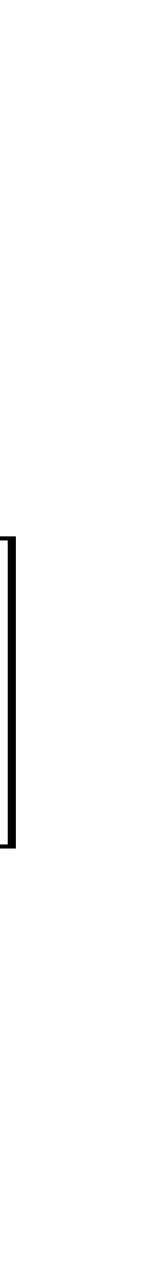
Let's verify this matrix does what its suppose to do:



Another Example: Scaling + Replacement



$R_3 \leftarrow (R_3 - 2R_1)$



Another Example: Scaling + Replacement Let's build the transformation:

Another Example: Scaling + Replacement Let's build the matrix which implements it:

Another Example: Scaling + Replacement Let's verify it does what it's suppose to do:

Elementary row operations are linear, so they are implemented by matrices

General Elementary Scaling Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

General Elementary Scaling Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

If we want to perform $R_3 \leftarrow kR_3$ then we need the identity matrix but with the entry $A_{33} = k$.

General Elementary Scaling Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

If we want to perform $R_3 \leftarrow kR_3$ then we need the identity matrix but with the entry $A_{33} = k$.

If we want to perform $R_i \leftarrow kR_i$ then we need the identity matrix but with then entry $A_{ii} = k$.

General Scaling + Replacement Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & 0 & 0 & 1 \end{bmatrix}$

General Scaling + Replacement Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & 0 & 0 & 1 \end{bmatrix}$

If we want to perform $R_4 \leftarrow R_4 + kR_1$, then we need the identity matrix but with the entry $A_{41} = k_{\bullet}$

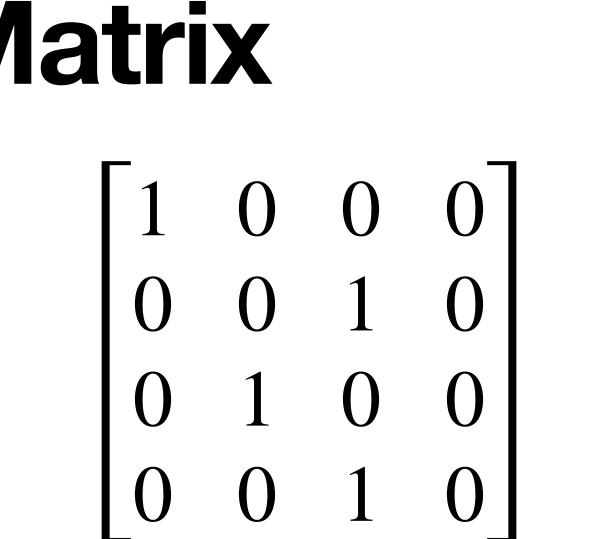
General Scaling + Replacement Matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & 0 & 0 & 1 \end{bmatrix}$

If we want to perform $R_i \leftarrow R_i + kR_j$, then we need the identity matrix but with the entry $A_{ij} = k$.

If we want to perform $R_4 \leftarrow R_4 + kR_1$, then we need the identity matrix but with the entry $A_{41} = k_{\bullet}$

General Swap Matrix

identity matrix, but with R_2 and R_3 swapped.



If we want to swap R_2 and R_3 , then we need the

Elementary Matrices

Definition. An **elementary matrix** is a matrix the identity matrix I.

Example.



obtained by applying a single row operation to

Elementary Matrices

Definition. An **elementary matrix** is a matrix the identity matrix I.



obtained by applying a single row operation to

These are exactly the matrices we were just looking at.

Elementary Matrices and Row Operations

Fact. Any elementary read an elementary matrix.
Verify:

Fact. Any elementary row can be implemented by

How To: Finding Elementary Matrices

Question. Find the matrix implementing the elementary row operation op.

appropriate size.

- Solution. Apply op to the identity matrix of the

Taking stock:

Taking stock:

» Elementary matrices implement elementary row operations.

Taking stock:

» Elementary matrices implement elementary row operations.

» Remember that Matrix multiplication is transformation composition (i.e., do one then the other).

Taking stock:

- » Elementary matrices implement elementary row operations.
- » Remember that Matrix multiplication is other).
- as a product of elementary matrices.

transformation composition (i.e., do one then the

So we can implement <u>any</u> sequence of row operations

How to: Matrices implementing Row Operations

Question. Find the matrix implementing a sequence of row operations op_1 , op_2 , ...

- Solution. Apply the row operations in sequence to the identity matrix of the appropriate size.



Question

Find the matrix implementing the following sequence of elementary row operations on a $3 \times n$ matrix.

Then multiply it with the all-ones 3×3 matrix.

- $R_2 \leftarrow 3R_2$
- $R_1 \leftarrow R_1 + R_2$
- swap R_2 and R_3



 $\begin{bmatrix} 1 & 3 & 0 \\ 0 & 0 & 1 \\ 0 & 3 & 0 \end{bmatrix}$



Second Key Observation

Second Key Observation

Elementary row operations.

Elementary row operations are **invertible** linear

Second Key Observation

transformations.

This also means the product of elementary matrices is invertible.

Elementary row operations are **invertible** linear

$(E_1 E_2 E_3 E_4)^{-1} = E_4^{-1} E_3^{-1} E_3^{-1} E_2^{-1} E_1^{-1}$!! the order reverses !!

Question (Conceptual)

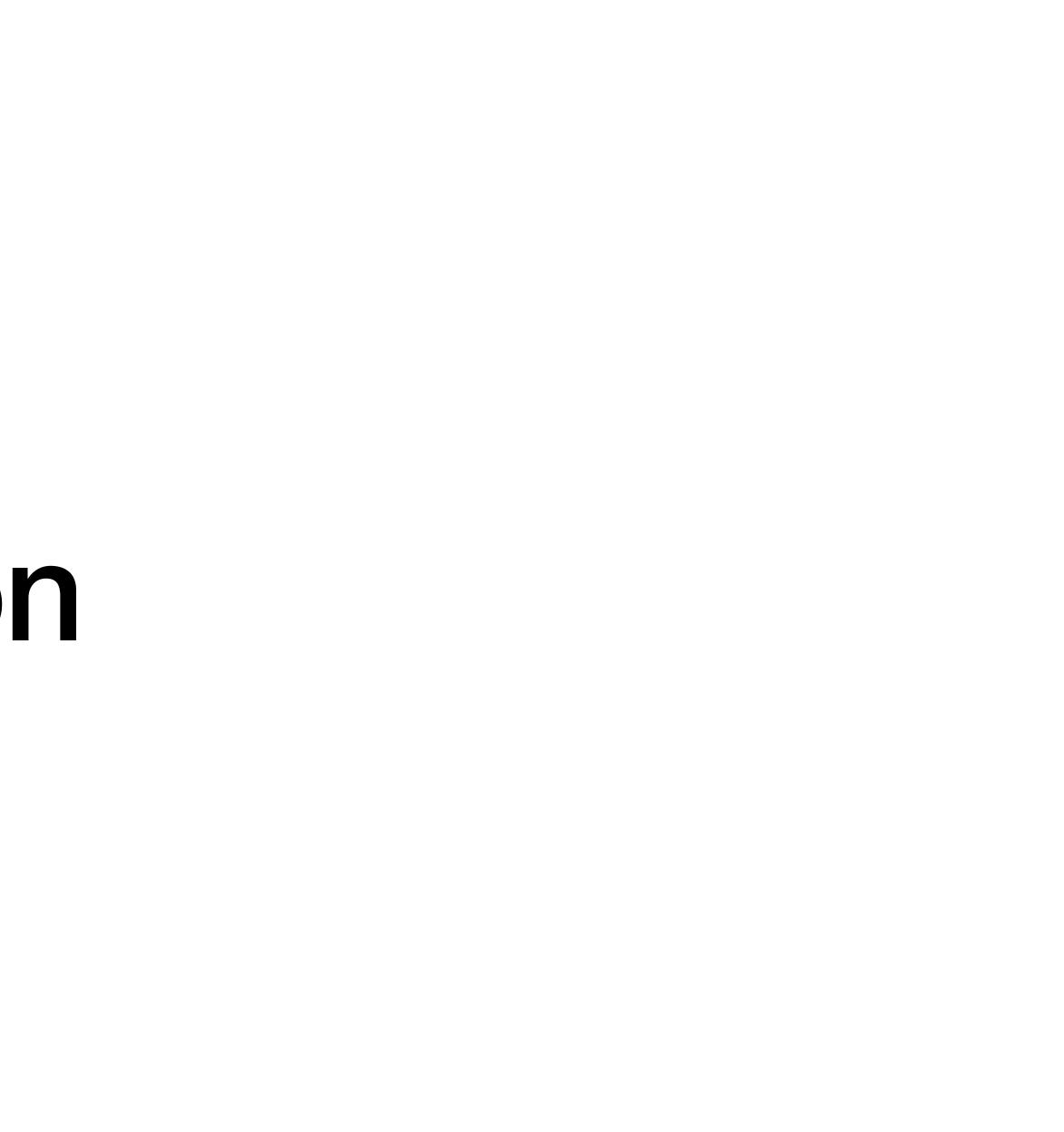
Describe the inverse transformation for each elementary row operation.



The inverse of $R_i \leftarrow R_i + R_i$ is $R_i \leftarrow R_i - R_i$. The inverse of swapping is swapping again.

- The inverse of scaling by k is scaling by 1/k.

LU Factorization



Recall: Elementary Row Operations

- scaling multiply a row by a number interchange switch two rows replacement add two rows (and replace one with the sum) rep. + scl. add a scaled equation to another

Recall: Elementary Row Operations We only need these two for the forward phase

interchange switch two rows

rep. + scl. add a scaled equation to another

A Simplifying Assumption

We'll assume for now we only need this one

rep. + scl. add a scaled equation to another

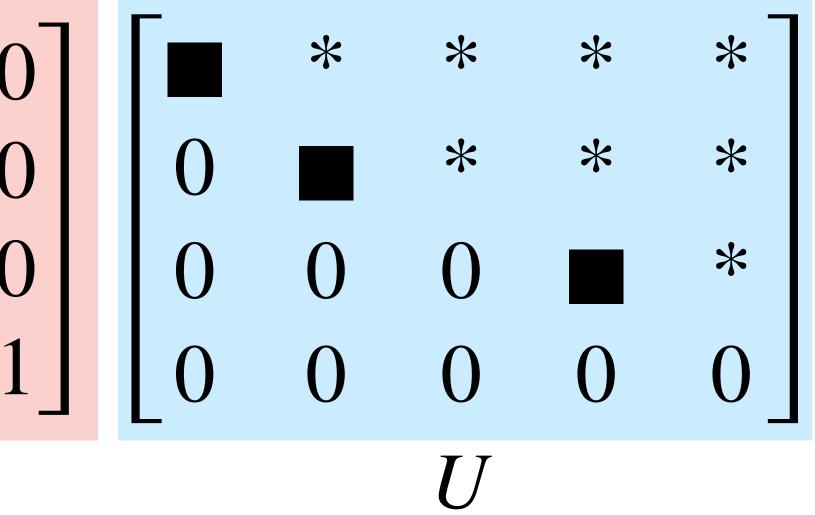
Reminder: LU Factorization at a High Level

Given a m×n matrix A, we are going to factorize A as

A

	1	0	0	C
=	*	1	0	C
	*	*	1	C
	*	*	*	1
		T		

Echelon form of A



1 FUNCTION LU_Factorization(A):

- **1 FUNCTION** LU_Factorization(*A*):
- 2 $L \leftarrow identity matrix$

- **1 FUNCTION** LU_Factorization(A):
- 2 $L \leftarrow identity matrix$
- $J \qquad U \leftarrow A$

- **FUNCTION** LU_Factorization(A): 1
- 2 $L \leftarrow identity matrix$
- $\mathsf{U} \leftarrow A$ 3
- 4

convert U to an echelon form by GE forward step # without swaps

- **FUNCTION** LU_Factorization(A): 1
- $L \leftarrow identity matrix$ 2
- $\mathsf{U} \leftarrow A$ 3
- 4
- **FOR** each row operation OP in the prev step: 5

convert U to an echelon form by GE forward step # without swaps

- **FUNCTION** LU_Factorization(A): 1
- $L \leftarrow identity matrix$ 2
- $U \leftarrow A$ 3
- 4
- FOR each row operation OP in the prev step: 5
- $E \leftarrow \text{the matrix implementing OP}$ 6

convert U to an echelon form by GE forward step # without swaps

FUNCTION LU_Factorization(A): 1 $L \leftarrow identity matrix$ 2 $\mathsf{U} \leftarrow A$ 3 4 **FOR** each row operation OP in the prev step: 5 6 $E \leftarrow \text{the matrix implementing OP}$ 7

convert U to an echelon form by GE forward step # without swaps

 $L \leftarrow L \oslash E^{-1}$ # note the multiplication on the right

FUNCTION LU_Factorization(A): 1 $L \leftarrow identity matrix$ 2 $\mathsf{U} \leftarrow A$ 3 4 **FOR** each row operation OP in the prev step: 5 6 $E \leftarrow \text{the matrix implementing OP}$ 7 RETURN (L, U) 8

convert U to an echelon form by GE forward step # without swaps

 $L \leftarrow L \oslash E^{-1}$ # note the multiplication on the right

FUNCTION LU_Factorization(A): 1 $L \leftarrow identity matrix$ 2 3 $\mathsf{U} \leftarrow A$ convert U to an echelon form by GE forward step # without swaps 4 **FOR** each row operation OP in the prev step: 5 $E \leftarrow \text{the matrix implementing OP}$ 6 $L \leftarrow L \oslash E^{-1}$ # note the multiplication on the right 7 RETURN (L, U) we'll see how to do this part smarter 8

S

Gaussian Elimination and Elementary Matrices

$A \sim A_1 \sim A_2 \sim \ldots \sim A_k$

Consider a sequence of elementary row operations from A to an echelon form.

elementary matrix.

Each step can be represent as a product with an

Gaussian Elimination and Elementary Matrices $A \rightarrow E A \rightarrow E E A$

 $A \sim E_1 A \sim E_2 E_1 A \sim \dots \sim E_k E_{k-1} \dots E_2 E_1 A$

Gaussian Elimination and Elementary Matrices $A \sim E_1 A \sim E_2 E_1 A \sim \dots \sim E_k E_{k-1} \dots E_2 E_1 A$

This exactly tells us that if B is the final echelon form we get then

where E implements a <u>sequence</u> of row operations.

 $B = (E_k E_{k-1} \dots E_2 E_1)A = EA$

Gaussian Elimination and Elementary Matrices $A \sim E_1 A \sim E_2 E_1 A \sim \ldots$

This exactly tells us that if B is the final echelon form we get then Invertible $B = (E_k E_{k-1} \dots E_2 E_1)A = EA$

where E implements a <u>sequence</u> of row operations.

$$- \mathcal{E}_k \mathcal{E}_{k-1} \cdots \mathcal{E}_2 \mathcal{E}_1 A$$

Gaussian Elimination and Elementary Matrices $A \sim E_1 A \sim E_2 E_1 A \sim \ldots$

This exactly tells us that if B is the final echelon form we get then Invertible $B = (E_k E_{k-1} \dots E_2 E_1)A = EA$ where E implements a <u>sequence</u> of row operations.

So

$$- \mathcal{E}_k \mathcal{E}_{k-1} \cdots \mathcal{E}_2 \mathcal{E}_1 A$$

 $A = E^{-1}B = (E_1^{-1}E_2^{-1}\dots E_{k-1}^{-1}E_k^{-1})B$

A New Perspective on Gaussian Elimination

The forward part of Gaussian elimination <u>is</u> matrix factorization

The "L" Part $E = E_k E_{k-1} \dots E_2 E_1$ This a product of elementary matrices So $L = E^{-1} = E_1^{-1}E_2^{-1}\dots E_{k-1}^{-1}E_k^{-1}$!! the order reverses !! We won't prove this, but it's worth thinking about: why is this lower triangular? And can we build this in a more efficient way?



demo

How To: LU Factorization by hand

- Question. Find a LU Factorization for the matrix A (assuming no swaps). Solution.
- » Start with L as the identity matrix. » Find U by the forward part of GE.

» For each operation $R_i \leftarrow R_i + kR_j$, set L_{ij} to -k.

Solving Systems using the LU Factorization

How To: Solving systems with the LU

A = LU is a LU factorization.

then solve $A\mathbf{x} = \mathbf{c}$ to get a solution d.

Verify:

- Question. Solve the equation $A\mathbf{x} = \mathbf{b}$ given that
- **Solution.** First solve Lx = b to get a solution c,

How To: Solving systems with the LU

A = LU is a LU factorization.

then solve $A\mathbf{x} = \mathbf{c}$ to get a solution d.

- Question. Solve the equation $A\mathbf{x} = \mathbf{b}$ given that
- **Solution.** First solve Lx = b to get a solution c,

Why is this better than just solving Ax = b?

FLOPs for Solving General Systems

<u>The following FLOP estimates are based on $n \times n$ matrices</u> Gaussian Elimination: $\sim \frac{2n^3}{3}$ FLOPS GE Forward: $\sim \frac{2n^3}{3}$ FLOPS GE Backward: $\sim 2n^2$ FLOPS Matrix Inversion: $\sim 2n^3$ FLOPS Matrix-Vector Multiplication: $\sim 2n^2$ FLOPS **Solving by matrix inversion:** $\sim 2n^3$ FLOPS **Solving by Gaussian elimination:** $\sim \frac{2n^3}{3}$ FLOPS

FLOPS for solving LU systems

LU Factorization: $\sim \frac{2n^3}{3}$ FLOPS Solving by LU Factorizati

- Solving $L\mathbf{x} = \mathbf{b}$: ~ $2n^2$ FLOPS (by "forward" elimination)
- Solving $U\mathbf{x} = \mathbf{c}$: ~ $2n^2$ FLOPS (already in echelon form)

Lon:
$$\sim \frac{2n^3}{3}$$
 FLOPS

If you solve several matrix equations for the same matrix, LU factorization is <u>faster</u> than matrix inversion on the *first* equation, and the same (in the worst case) in later equation.

If A doesn't have to many entries (A is sparse), then its likely that L and U won't either.

If A doesn't have to many entries (A is sparse), then its likely that L and U won't either.

But A^{-1} may have *many* entries (A^{-1} is dense)

If A doesn't have to many entries (A is sparse), then its likely that L and U won't either.

But A^{-1} may have *many* entries (A^{-1} is dense)

better with respect to storage.

Sparse matrices are faster to compute with and

Summary

We can factorize matrices to make them easier to work with, or get more information about them

LU Factorizations allow us to solve multiple matrix equations, with one forward step and multiple backwards steps.