Matrix-vector products

The operational way: row-by-column $b_i = \sum_j A_{ij} c_j$.

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

The smart way: linear combinations of columns of A

$$\underbrace{\begin{bmatrix} A_{11} \\ A_{21} \\ A_{31} \\ A_{41} \end{bmatrix}}_{v_1} c_1 + \underbrace{\begin{bmatrix} A_{12} \\ A_{22} \\ A_{32} \\ A_{42} \end{bmatrix}}_{v_2} c_2 + \underbrace{\begin{bmatrix} A_{31} \\ A_{32} \\ A_{33} \\ A_{43} \end{bmatrix}}_{v_3} c_3 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

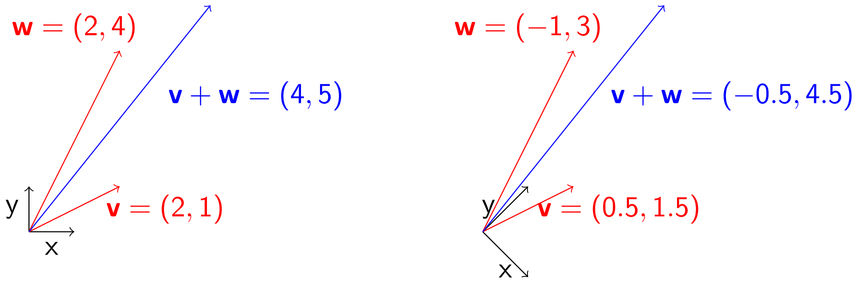
The entries of $\mathbf{c}$ are coordinates used to write $\mathbf{b}$ as a linear combination of v_1, v_2, v_3.

Linear algebra in a slide

(You have already seen linear algebra, right?)

The powerful idea behind linear algebra

- there is a 'space' of vectors as abstract geometrical objects.
- operations on vectors $\iff$ operations on coordinates.
- many relations are true regardless of the choice of coordinates.



Bases

Once we have fixed basis $v_1, v_2, \dots, v_m$, we can write vectors as coordinates wrt them.

Canonical basis: vectors with only one 1; e.g. for $m = 4$

$$\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \quad \mathbf{e}_4 = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

Coordinates wrt this basis $\iff$ vector entries

$\mathbf{b} = \mathbf{e}_1 b_1 + \mathbf{e}_2 b_2 + \mathbf{e}_3 b_3 + \mathbf{e}_4 b_4$.

(I like to put scalars on the right.)

(In real life, vectors are not always boldfaced/underlined for your convenience.)

Linear systems

Problem: find coordinates $x_1, \dots, x_n$ needed to write b as linear combinations of the columns of $A \in \mathbb{R}^{m \times n}$, or

$$Ax = b.$$

Sometimes there are multiple solutions, or none, e.g.,

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 0 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 4 \\ 4 \\ 1 \\ 2 \end{bmatrix}.$$

Im A: set of vectors b that we can obtain.
ker A: possible choices of x that produce zero.

Main problem that we will treat in this module: find x that reaches a given b exactly, or gets as close as possible.

Square linear systems

A is called invertible if it is square and each vector is reachable.

In this case, the solution is given by another matrix: $x = A^{-1}b$

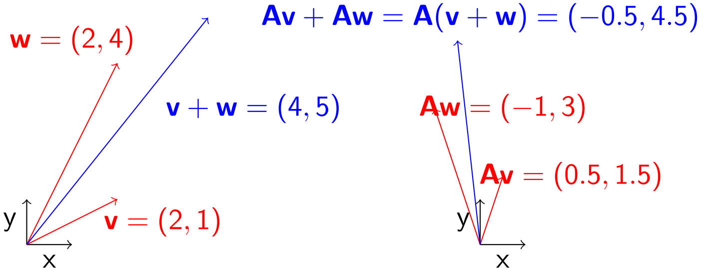
$$AA^{-1} = A^{-1}A = I = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

(Convention: omitted elements are zero.)

Warning `inv(A)*b` is not the best way to solve $Ax = b$, numerically.
Most languages have a specialized instruction, e.g., Matlab's `x = A \ b` or Python's `scipy.linalg.solve(A, b)`.

Matrices as transformations

The other idea behind linear algebra: matrices represent linear transformations of the space, e.g.,



I represents the identity, i.e., $Av = v$ for each v.

$A(Bv) =$ "apply B first, then A" is another transformation, represented by the matrix AB (so that $A(Bv) = (AB)v$).

Matrix-matrix product

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix}$$

$A \in \mathbb{R}^{4\times 3}$, $B \in \mathbb{R}^{3\times 2}$. $AB \in \mathbb{R}^{4\times 2}$.

Mnemonic: if the inner dimensions agree, the product is well-defined and removes them.

We can identify vectors with columns ($n \times 1$ matrices).

Cost: multiplying $m \times n$ and $n \times p$ requires $m(2n-1)p$ floating point operations (flops). Forget about fancier algorithms (e.g. Strassen).

Slightly different beast: number-times-matrix, e.g.

$$3A = \begin{bmatrix} 3A_{11} & 3A_{12} & \dots \\ 3A_{21} & 3A_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Order of operations

Usual algebra properties hold, e.g.: $A(B+C) = AB + AC$, $A(BC) = (AB)C$, etc.

Warning: Parenthesization matters a lot: if $A, B \in \mathbb{R}^{n \times n}, v \in \mathbb{R}^n$, then $(AB)v$ costs $O(n^3)$, but $A(Bv)$ costs $O(n^2)$.

Matlab example:

```
n = 2000;
A = randn(n, n);
B = randn(n, n);
v = randn(n, 1);
tic, A * (B * v); toc
tic, (A * B) * v; toc
```

Warning: programming languages usually do not rearrange parentheses to help you.

Matrix algebra

$$(A+B)^2 = (A+B)(A+B) = A^2 + AB + BA + B^2.$$

What doesn't work

$AB \neq BA$: might not even make sense dimension-wise.
Exception: We can move around numbers (scalars): $3AB = A(3B)$.

$AB = AC$ does not imply $B = C$ (example).

However, if there is a matrix M such that $MA = I$, I can multiply by M:

$$(MA)B = (MA)C \iff B = C.$$

Warning: multiplying 'on the left' and 'on the right' differ.

Row and column vectors

$$v = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}, v^T = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}.$$

v is a vector in $\mathbb{R}^3$ (or a matrix in $\mathbb{R}^{3\times1}$). v^T is a matrix in $\mathbb{R}^{1\times3}$ (or row vector).

```
>> v = [4;5;6]
v =
     4
     5
     6
>> w = [1 2 3]
w =
     1 2 3
>> w*v
ans =
    32
```

Row and column vectors

```
>> v*w
ans =
    4 8 12
    5 10 15
    6 12 18
>> v'
ans =
    4 5 6
>> w*v'
Error using *
Inner matrix dimensions must agree.
```

Some people (even other professors) write vw when they mean $v^T w$. This will be confusing (what is uvw?).

$v \cdot w$: more acceptable; at least it's clear it is a different operation.

Block operations

When computing a matrix product, we get the same result if we use the row-by-column rule block-wise.

$$
\left[\begin{array}{ccc|c|cc}
* & * & * & * & * & * \\
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\
* & * & * & * & * & *
\end{array}\right]
\cdot
\left[\begin{array}{ccc|cc|c}
* & * & * & * & * & * \\
* & * & * & * & * & * \\
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\
* & * & * & * & * & *
\end{array}\right]
=
\left[\begin{array}{ccc|cc|c}
* & * & * & * & * & * \\
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\
* & * & * & * & * & * \\ \hline
* & * & * & * & * & * \\
* & * & * & * & * & *
\end{array}\right]
$$

$$
\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix}
\cdot
\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}
+
\begin{bmatrix} * \\ * \end{bmatrix}
\cdot
\begin{bmatrix} * & * & * \end{bmatrix}
+
\begin{bmatrix} * & * \\ * & * \end{bmatrix}
\cdot
\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix}
=
\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix}
$$

In $AB = C$, columns of A and rows of B must be partitioned in the same way, for the product to make sense.
(Matlab example — syntax `A(1:2, 1:3)`.)

Block operations

When implementing linear algebra on a computer, usually chopping up matrices into large blocks gives better performance (even with an equal number of floating point operations), because of caching/locality reasons.

This is one of the reasons why library calls usually perform better than hand-coded loops.

Dividing matrices into blocks is useful also for the analysis of algorithms and operations: for instance, block triangular matrices are closed under products:

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix} \begin{bmatrix} D & E \\ 0 & F \end{bmatrix} = \begin{bmatrix} AD & AE + BF \\ 0 & CF \end{bmatrix}. \tag{*}$$

(0 here stands for a block of zeros.)

Block triangular matrices

Let $A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1k} \\ 0 & A_{22} & \dots & A_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{kk} \end{bmatrix}$ be a block triangular matrix, with all A_{ii} square.

- The product of two block (upper/lower) triangular matrices (with compatible block sizes) is still block triangular — see (*) in previous slide.
- A block triangular matrix is invertible iff all diagonal blocks A_{ii} are invertible
- Its eigenvalues are the union of the eigenvalues of the A_{ii}.

(Matlab example: compute eigenvalues with `eig`).

Example: 2×2 block triangular linear system

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}.$$

(Again, diagonal blocks are square and all dimensions are compatible.)

$$\begin{bmatrix} Ax + By \\ Cy \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} \implies y = C^{-1}f,\, x = A^{-1}(e - BC^{-1}f).$$

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}BC^{-1} \\ 0 & C^{-1} \end{bmatrix}.$$

(Informal idea: we can start solving from the variables in C.)

Cheaper than a general system solution.
General principle: matrix structures matter.

Exercises

1. Write down precisely the dimensions of all matrices in the previous example of a 2×2 block triangular linear system. Be careful — A and C may be square but of different dimensions here, for instance $A \in \mathbb{R}^{m \times m}$, $C \in \mathbb{R}^{n \times n}$.
2. What is the computational cost (up to lower order terms) of computing the product of two square matrices $A, B \in \mathbb{R}^{n \times n}$? Of a matrix-vector product Av, $v \in \mathbb{R}^n$?
3. What is the computational cost of solving a triangular linear system by back-substitution 'starting from the last equation'?
4. Let $A = I + uu^T$, where I is the $n \times n$ identity matrix (what is it?) and u is a vector. How can one compute the product Av (for a vector v) in $O(n)$ flops?

Exercises

1. Compute the product of two 3×3 block lower triangular matrices, i.e., two of the form

$$\begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

   (all A_{ij} here are square matrices, not numbers.) Be careful with the order of the factors.
2. Simplify the expression $A^{-1}(A-B)B^{-1}(A-B)$.
3. What is the inverse of a matrix of the form $\begin{bmatrix} 0 & A \\ B & C \end{bmatrix}$ (all blocks square of the same size)? Is the product of two matrices in this form still in the same form? (Suppose all blocks are square.)
4. Suppose that the adjacency matrix of a graph is block triangular. What does this imply on the graph?

References

Trefethen-Bau book, chapter 1 (matrix-vector product).

Other exercises (also more challenging) on the Trefethen-Bau and Demmel books.