3. Matrices

- notation and terminology
- matrix operations
- linear and affine functions
- complexity
Matrix

A rectangular array of numbers, for example

\[
A = \begin{bmatrix}
0 & 1 & -2.3 & 0.1 \\
1.3 & 4 & -0.1 & 0 \\
4.1 & -1 & 0 & 1.7
\end{bmatrix}
\]

- numbers in array are the elements (components, entries, coefficients)
- \(A_{ij}\) is the \(i, j\) element of \(A\); \(i\) is its row index, \(j\) the column index
- size of the matrix is (\#rows) \(\times\) (\#columns), e.g., \(A\) is a 3 \(\times\) 4 matrix
- set of \(m \times n\) matrices with real elements is written \(\mathbb{R}^{m \times n}\)
- set of \(m \times n\) matrices with complex elements is written \(\mathbb{C}^{m \times n}\)
Other conventions

• many authors use parentheses as delimiters:

\[ A = \begin{pmatrix}
0 & 1 & -2.3 & 0.1 \\
1.3 & 4 & -0.1 & 0 \\
4.1 & -1 & 0 & 1.7
\end{pmatrix} \]

• often \( a_{ij} \) is used to denote the \( i, j \) element of \( A \)
Matrix shapes

Scalar: we don’t distinguish between a $1 \times 1$ matrix and a scalar

Vector: we don’t distinguish between an $n \times 1$ matrix and an $n$-vector

Row and column vectors

- a $1 \times n$-matrix is called a row vector
- an $n \times 1$-matrix is called a column vector (or just vector)

Tall, wide, square matrices: an $m \times n$-matrix is

- tall if $m > n$
- wide if $m < n$
- square if $m = n$
Block matrix

- a block matrix is a rectangular array of matrices
- elements in the array are the blocks or submatrices of the block matrix

Example

\[
A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}
\]

is a $2 \times 2$ block matrix; if the blocks are

\[
B = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 2 & 3 \\ 5 & 4 & 7 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \end{bmatrix}, \quad E = \begin{bmatrix} -1 & 6 & 0 \end{bmatrix}
\]

then

\[
A = \begin{bmatrix} 2 & 0 & 2 & 3 \\ 1 & 5 & 4 & 7 \\ 1 & -1 & 6 & 0 \end{bmatrix}
\]

Note: dimensions of the blocks must be compatible!
Rows and columns

a matrix can be viewed as a block matrix with row/column vector blocks

- \( m \times n \) matrix \( A \) as \( 1 \times n \) block matrix

\[
A = \begin{bmatrix}
  a_1 & a_2 & \cdots & a_n
\end{bmatrix}
\]

each \( a_j \) is an \( m \)-vector (the \( j \)th column of \( A \))

- \( m \times n \) matrix \( A \) as \( m \times 1 \) block matrix

\[
A = 
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_m
\end{bmatrix}
\]

each \( b_i \) is a \( 1 \times n \) row vector (the \( i \)th row of \( A \))
Special matrices

Zero matrix

- matrix with $A_{ij} = 0$ for all $i, j$
- notation: 0 (usually) or $0_{m \times n}$ (if dimension is not clear from context)

Identity matrix

- square matrix with $A_{ij} = 1$ if $i = j$ and $A_{ij} = 0$ if $i \neq j$
- notation: $I$ (usually) or $I_n$ (if dimension is not clear from context)
- columns of $I_n$ are unit vectors $e_1, e_2, \ldots, e_n$; for example,

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$$
Symmetric and Hermitian matrices

**Symmetric matrix:** square with $A_{ij} = A_{ji}$

\[
\begin{bmatrix}
4 & 3 & -2 \\
3 & -1 & 5 \\
-2 & 5 & 0 \\
\end{bmatrix}
, 
\begin{bmatrix}
4 + 3j & 3 - 2j & 0 \\
3 - 2j & -j & -2j \\
0 & -2j & 3 \\
\end{bmatrix}
\]

**Hermitian matrix:** square with $A_{ij} = \bar{A}_{ji}$ (complex conjugate of $A_{ij}$)

\[
\begin{bmatrix}
4 & 3 - 2j & -1 + j \\
3 + 2j & -1 & 2j \\
-1 - j & -2j & 3 \\
\end{bmatrix}
\]

note: diagonal elements are real (since $A_{ii} = \bar{A}_{ii}$)
Structured matrices

matrices with special patterns or structure arise in many applications

- diagonal matrix: square with $A_{ij} = 0$ for $i \neq j$

\[
\begin{bmatrix}
-1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & -5 \\
\end{bmatrix}, \quad
\begin{bmatrix}
-1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & -5 \\
\end{bmatrix}
\]

- lower triangular matrix: square with $A_{ij} = 0$ for $i < j$

\[
\begin{bmatrix}
4 & 0 & 0 \\
3 & -1 & 0 \\
0 & 5 & -2 \\
\end{bmatrix}
\]

- upper triangular matrix: square with $A_{ij} = 0$ for $i > j$
Sparse matrices

A matrix is *sparse* if most (almost all) of its elements are zero.

- Sparse matrix storage formats and algorithms exploit sparsity.
- Efficiency depends on the number of nonzeros and their positions.
- Positions of nonzeros are visualized in a ‘spy plot’.

**Example**

- 2,987,012 rows and columns
- 26,621,983 nonzeros

(Freescale/FullChip matrix from Univ. of Florida Sparse Matrix Collection)
Outline

• notation and terminology

• **matrix operations**

• linear and affine functions

• complexity
Scalar-matrix multiplication and addition

Scalar-matrix multiplication:

scalar-matrix product of $m \times n$ matrix $A$ with scalar $\beta$

$$\beta A = \begin{bmatrix} \beta A_{11} & \beta A_{12} & \cdots & \beta A_{1n} \\ \beta A_{21} & \beta A_{22} & \cdots & \beta A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta A_{m1} & \beta A_{m2} & \cdots & \beta A_{mn} \end{bmatrix}$$

$A$ and $\beta$ can be real or complex

Addition: sum of two $m \times n$ matrices $A$ and $B$ (real or complex)

$$A + B = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} & \cdots & A_{1n} + B_{1n} \\ A_{21} + B_{21} & A_{22} + B_{22} & \cdots & A_{2n} + B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} + B_{m1} & A_{m2} + B_{m2} & \cdots & A_{mn} + B_{mn} \end{bmatrix}$$
Transpose

the *transpose* of an $m \times n$ matrix $A$ is the $n \times m$ matrix

$$A^T = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} \\ A_{12} & A_{22} & \cdots & A_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{mn} \end{bmatrix}$$

- $(A^T)^T = A$

- a symmetric matrix satisfies $A = A^T$

- $A$ may be complex, but transpose of complex matrices is rarely needed

- transpose of matrix-scalar product and matrix sum

$$\left(\beta A\right)^T = \beta A^T, \quad (A + B)^T = A^T + B^T$$
Conjugate transpose

the *conjugate transpose* of an $m \times n$ matrix $A$ is the $n \times m$ matrix

$$A^H = \begin{bmatrix}
\bar{A}_{11} & \bar{A}_{21} & \cdots & \bar{A}_{m1} \\
\bar{A}_{12} & \bar{A}_{22} & \cdots & \bar{A}_{m2} \\
\vdots & \vdots & \ddots & \vdots \\
\bar{A}_{1n} & \bar{A}_{2n} & \cdots & \bar{A}_{mn}
\end{bmatrix}$$

($\bar{A}_{ij}$ is complex conjugate of $A_{ij}$)

- $A^H = A^T$ if $A$ is a real matrix
- a Hermitian matrix satisfies $A = A^H$
- conjugate transpose of matrix-scalar product and matrix sum

$$(\beta A)^H = \bar{\beta} A^H, \quad (A + B)^H = A^H + B^H$$
Matrix-matrix product

product of $m \times n$ matrix $A$ and $n \times p$ matrix $B$ ($A$, $B$ real or complex)

$$C = AB$$

is the $m \times p$ matrix with elements

$$C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j} + \cdots + A_{in}B_{nj}$$

dimensions must be compatible: #columns in $A$ = #rows in $B$
Exercise: paths in directed graph

directed graph with $n = 5$ vertices

matrix representation

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}$$

$A_{ij} = 1$ indicates an edge $j \rightarrow i$

Question: give a graph interpretation of $A^2 = AA$, $A^3 = AAA$, …

$$A^2 = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 2 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
\end{bmatrix}, \quad A^3 = \begin{bmatrix}
1 & 1 & 0 & 1 & 2 \\
2 & 0 & 1 & 2 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}, \quad …$$
Properties of matrix-matrix product

- **associative:** \((AB)C = A(BC)\) so we write \(ABC\)

- **associative with scalar-matrix multiplication:** \((\gamma A)B = \gamma(AB) = \gamma AB\)

- **distributes with sum:**

  \[
  A(B + C) = AB + AC, \quad (A + B)C = AC + BC
  \]

- **transpose and conjugate transpose of product:**

  \[
  (AB)^T = B^T A^T, \quad (AB)^H = B^H A^H
  \]

- **not commutative:** \(AB \neq BA\) in general; for example,

  \[
  \begin{bmatrix}
  -1 & 0 \\
  0 & 1
  \end{bmatrix}
  \begin{bmatrix}
  0 & 1 \\
  1 & 0
  \end{bmatrix}
  \neq
  \begin{bmatrix}
  0 & 1 \\
  1 & 0
  \end{bmatrix}
  \begin{bmatrix}
  -1 & 0 \\
  0 & 1
  \end{bmatrix}
  \]

  there are exceptions, e.g., \(AI = IA\) for square \(A\)
Notation for vector inner product

• inner product of \(a, b \in \mathbb{R}^n\) (see page 1-18):

\[
b^T a = b_1 a_1 + b_2 a_2 + \cdots + b_n a_n =
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_n 
\end{bmatrix}^T
\begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_n 
\end{bmatrix}
\]

product of the transpose of column vector \(b\) and column vector \(a\)

• inner product of \(a, b \in \mathbb{C}^n\) (see page 1-25):

\[
b^H a = \bar{b}_1 a_1 + \bar{b}_2 a_2 + \cdots + \bar{b}_n a_n =
\begin{bmatrix}
    \bar{b}_1 \\
    \bar{b}_2 \\
    \vdots \\
    \bar{b}_n 
\end{bmatrix}^H
\begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_n 
\end{bmatrix}
\]

product of conjugate transpose of column vector \(b\) and column vector \(a\)
Matrix-matrix product and block matrices

block-matrices can be multiplied as regular matrices

**Example:** product of two $2 \times 2$ block matrices

\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix}
\begin{bmatrix}
W & Y \\
X & Z
\end{bmatrix}
= \begin{bmatrix}
AW + BX & AY + BZ \\
CW + DX & CY + DZ
\end{bmatrix}
\]

if the dimensions of the blocks are compatible
Outline

• notation and terminology
• matrix operations
• linear and affine functions
• complexity
Matrix-vector product

product of $m \times n$ matrix $A$ with $n$-vector (or $n \times 1$ matrix) $x$

$$Ax = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix}$$

- dimensions must be compatible: #columns of $A = \text{length of } x$
- $Ax$ is a linear combination of the columns of $A$:

$$Ax = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1a_1 + x_2a_2 + \cdots + x_na_n$$

each $a_i$ is an $m$-vector ($i$th column of $A$)
Linear function

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is linear if superposition holds:

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

for all $n$-vectors $x, y$ and all scalars $\alpha, \beta$

Extension: if $f$ is linear, superposition holds for any linear combination:

$$f(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_p u_p) = \alpha_1 f(u_1) + \alpha_2 f(u_2) + \cdots + \alpha_p f(u_p)$$

for all scalars, $\alpha_1, \ldots, \alpha_p$ and all $n$-vectors $u_1, \ldots, u_p$
Matrix-vector product function

for fixed $A \in \mathbb{R}^{m \times n}$, define a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$f(x) = Ax$$

• any function of this type is linear: $A(\alpha x + \beta y) = \alpha(Ax) + \beta(Ay)$

• every linear function can be written as a matrix-vector product function:

$$f(x) = f(x_1e_1 + x_2e_2 + \cdots + x_ne_n) = x_1f(e_1) + x_2f(e_2) + \cdots + x_nf(e_n)$$

$$= \begin{bmatrix} f(e_1) & \cdots & f(e_n) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

hence, $f(x) = Ax$ with $A = \begin{bmatrix} f(e_1) & f(e_2) & \cdots & f(e_n) \end{bmatrix}$
Input-output (operator) interpretation

think of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in terms of its effect on $x$

\[
\begin{array}{c}
x \xrightarrow{A} y = f(x) = Ax
\end{array}
\]

- signal processing/control interpretation: $n$ inputs $x_i$, $m$ outputs $y_i$
- $f$ is linear if we can represent its action on $x$ as a product $f(x) = Ax$
Examples \( (f : \mathbb{R}^3 \rightarrow \mathbb{R}^3) \)

- \( f \) reverses the order of the components of \( x \)
  
  a linear function: \( f(x) = Ax \) with
  
  \[
  A = \begin{bmatrix}
  0 & 0 & 1 \\
  0 & 1 & 0 \\
  1 & 0 & 0
  \end{bmatrix}
  \]

- \( f \) sorts the components of \( x \) in decreasing order: not linear

- \( f \) scales \( x_1 \) by a given number \( d_1 \), \( x_2 \) by \( d_2 \), \( x_3 \) by \( d_3 \)
  
  a linear function: \( f(x) = Ax \) with
  
  \[
  A = \begin{bmatrix}
  d_1 & 0 & 0 \\
  0 & d_2 & 0 \\
  0 & 0 & d_3
  \end{bmatrix}
  \]

- \( f \) replaces each \( x_i \) by its absolute value \( |x_i| \): not linear
Operator interpretation of matrix-matrix product explains why in general $AB \neq BA$

Example

\[ A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \]

- $f(x) = ABx$ swaps the elements of $x$; then changes sign of first element
- $f(x) = BAx$ changes sign of 1st element; then swaps the two elements
Reversal and circular shift

Reversal matrix

\[ A = \begin{bmatrix}
0 & 0 & \cdots & 0 & 1 \\
0 & 0 & \cdots & 1 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 1 & \cdots & 0 & 0 \\
1 & 0 & \cdots & 0 & 0
\end{bmatrix}, \quad Ax = (x_n, x_{n-1}, \ldots, x_2, x_1) \]

Circular shift matrix

\[ A = \begin{bmatrix}
0 & 0 & \cdots & 0 & 1 \\
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0
\end{bmatrix}, \quad Ax = (x_n, x_1, x_2, \ldots, x_{n-1}) \]
Permutation

Permutation matrix

- a square 0-1 matrix with one element 1 per row and one 1 per column
- equivalently, an identity matrix with columns reordered
- equivalently, an identity matrix with rows reordered

$Ax$ is a permutation of the elements of $x$

Example

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad Ax = (x_2, x_4, x_1, x_3)$$
Rotation in a plane

\[ A = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \]

\( Ax \) is \( x \) rotated counterclockwise over an angle \( \theta \)
Projection on line and reflection

- projection on line through \( a \) (see page 2-11):

\[
y = \frac{a^T x}{\|a\|^2} a = Ax \quad \text{with} \quad A = \frac{1}{\|a\|^2} aa^T
\]

- reflection with respect to line through \( a \)

\[
z = x + 2(y - x) = Bx, \quad \text{with} \quad B = \frac{2}{\|a\|^2} aa^T - I
\]
Node-arc incidence matrix

- directed graph with $m$ vertices, $n$ arcs (directed edges)
- incidence matrix is $m \times n$ matrix $A$ with

$$A_{ij} = \begin{cases} 
1 & \text{if arc } j \text{ enters node } i \\
-1 & \text{if arc } j \text{ leaves node } i \\
0 & \text{otherwise}
\end{cases}$$

$$A = \begin{bmatrix}
-1 & -1 & 0 & 1 & 0 \\
1 & 0 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 & -1 \\
0 & 1 & 0 & 0 & 1
\end{bmatrix}$$
Kirchhoff’s current law

$n$-vector $x = (x_1, x_2, \ldots, x_n)$ with $x_j$ the current through arc $j$

$$(Ax)_i = \sum_{\text{arc } j \text{ enters node } i} x_j - \sum_{\text{arc } j \text{ leaves node } i} x_j$$

$= \text{total current arriving at node } i$

\[
\begin{bmatrix}
-x_1 - x_2 + x_4 \\
x_1 - x_3 \\
x_3 - x_4 - x_5 \\
x_2 + x_5
\end{bmatrix}
\]
Kirchhoff’s voltage law

$m$-vector $y = (y_1, y_2, \ldots, y_m)$ with $y_i$ the potential at node $i$

$$(A^T y)_j = y_k - y_l \quad \text{if edge } j \text{ goes from node } l \text{ to } k$$

$= \text{ negative of voltage across arc } j$

Matrices 3-31
Vandermonde matrix

- polynomial of degree $n - 1$ or less with coefficients $x_1, x_2, \ldots, x_n$:

$$p(t) = x_1 + x_2t + x_3t^2 + \cdots + x_n t^{n-1}$$

- values of $p(t)$ at $m$ points $t_1, \ldots, t_m$:

$$\begin{bmatrix}
p(t_1) \\
p(t_2) \\
\vdots \\
p(t_m)
\end{bmatrix} = 
\begin{bmatrix}
1 & t_1 & \cdots & t_1^{n-1} \\
1 & t_2 & \cdots & t_2^{n-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & t_m & \cdots & t_m^{n-1}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix} = Ax$$

the matrix $A$ is called a Vandermonde matrix

- $f(x) = Ax$ maps coefficients of polynomial to function values
Discrete Fourier transform

the DFT maps a complex \( n \)-vector \((x_1, x_2, \ldots, x_n)\) to the complex \( n \)-vector

\[
\begin{bmatrix}
  y_1 \\
y_2 \\
y_3 \\
\vdots \\
y_n
\end{bmatrix}
= \begin{bmatrix}
  1 & 1 & 1 & \cdots & 1 \\
  1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\
  1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_n
\end{bmatrix}
= W \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_n
\end{bmatrix}
\]

where \( \omega = e^{2\pi j/n} \) (and \( j = \sqrt{-1} \))

- DFT matrix \( W \in \mathbb{C}^{n \times n} \) has \( k, l \) element \( W_{kl} = \omega^{-(k-1)(l-1)} \)
- a Vandermonde matrix with \( m = n \) and

\[
t_1 = 1, \quad t_2 = \omega^{-1}, \quad t_3 = \omega^{-2}, \quad \ldots, \quad t_n = \omega^{-(n-1)}
\]
Affine function

A function \( f : \mathbb{R}^n \to \mathbb{R}^m \) is affine if it satisfies

\[
f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)
\]

for all \( n \)-vectors \( x, y \) and all scalars \( \alpha, \beta \) with \( \alpha + \beta = 1 \)

**Extension:** if \( f \) is affine, then

\[
f(\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m) = \alpha_1 f(u_1) + \alpha_2 f(u_2) + \cdots + \alpha_m f(u_m)
\]

for all \( n \)-vectors \( u_1, \ldots, u_m \) and all scalars \( \alpha_1, \ldots, \alpha_m \) with

\[
\alpha_1 + \alpha_2 + \cdots + \alpha_m = 1
\]
Affine functions and matrix-vector product

for fixed $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, define a function $f : \mathbb{R}^n \to \mathbb{R}^m$ by

$$f(x) = Ax + b$$

i.e., a matrix-vector product plus a constant

- any function of this type is affine: if $\alpha + \beta = 1$ then

$$A(\alpha x + \beta y) + b = \alpha(Ax + b) + \beta(Ax + b)$$

- every affine function can be written as $f(x) = Ax + b$ with:

$$A = \begin{bmatrix} f(e_1) - f(0) & f(e_2) - f(0) & \cdots & f(e_n) - f(0) \end{bmatrix}$$

and $b = f(0)$
Affine approximation

first-order Taylor approximation of differentiable \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) around \( z \):

\[
\hat{f}_i(x) = f_i(z) + \frac{\partial f_i}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f_i}{\partial x_n}(z)(x_n - z_n), \quad i = 1, \ldots, m
\]

in matrix-vector notation: \( \hat{f}(x) = f(z) + Df(z)(x - z) \) where

\[
Df(z) = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1}(z) & \frac{\partial f_1}{\partial x_2}(z) & \cdots & \frac{\partial f_1}{\partial x_n}(z) \\
\frac{\partial f_2}{\partial x_1}(z) & \frac{\partial f_2}{\partial x_2}(z) & \cdots & \frac{\partial f_2}{\partial x_n}(z) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1}(z) & \frac{\partial f_m}{\partial x_2}(z) & \cdots & \frac{\partial f_m}{\partial x_n}(z)
\end{bmatrix}
= \begin{bmatrix}
\nabla f_1(z)^T \\
\nabla f_2(z)^T \\
\vdots \\
\nabla f_m(z)^T
\end{bmatrix}
\]

- \( Df(z) \) is called the derivative matrix or Jacobian matrix of \( f \) at \( z \)
- \( \hat{f} \) is a local affine approximation of \( f \) around \( z \)
Example

\[ f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} e^{2x_1 + x_2} - x_1 \\ x_1^2 - x_2 \end{bmatrix} \]

• derivative matrix

\[ Df(x) = \begin{bmatrix} 2e^{2x_1 + x_2} - 1 & e^{2x_1 + x_2} \\ 2x_1 & -1 \end{bmatrix} \]

• first order approximation of \( f \) around \( z = 0 \):

\[ \hat{f}(x) = \begin{bmatrix} \hat{f}_1(x) \\ \hat{f}_2(x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \]
Outline

• notation and terminology

• matrix operations

• linear and affine functions

• complexity
Matrix-vector product

Matrix-vector multiplication of \( m \times n \) matrix \( A \) and \( n \)-vector \( x \):

\[
y = Ax
\]

requires \((2n - 1)m\) flops

- \( m \) elements in \( y \); each element requires an inner product of length \( n \)
- approximately \( 2mn \) for large \( n \)

**Special cases:** flop count is lower for structured matrices

- \( A \) diagonal: \( n \) flops
- \( A \) lower triangular: \( n^2 \) flops
- \( A \) sparse: \( \# \)flops \( \ll 2mn \)
Matrix-matrix product

product of $m \times n$ matrix $A$ and $n \times p$ matrix $B$:

$$C = AB$$

requires $mp(2n - 1)$ flops

- $mp$ elements in $C$; each element requires an inner product of length $n$
- approximately $2mnp$ for large $n$
Exercises

1. evaluate $y = ABx$ two ways ($A$ and $B$ are $n \times n$, $x$ is a vector)
   
   • $y = (AB)x$ (first make product $C = AB$, then multiply $C$ with $x$)
   • $y = A(Bx)$ (first make product $y = Bx$, then multiply $A$ with $y$)
   
   both methods give the same answer, but which method is faster?

2. evaluate $y = (I + uv^T)x$ where $u$, $v$, $x$ are $n$-vectors
   
   • $A = I + uv^T$ followed by $y = Ax$
     
     in MATLAB: $y = (\text{eye}(n) + u*v') \times x$
   
   • $w = (v^T x)u$ followed by $y = x + w$
     
     in MATLAB: $y = x + (v' * x) \times u$