Numerical Linear Algebra

Review

Gerard Sleijpen and Martin van Gijzen

September 27, 2017

Instructor & Co-author

Dr. Gerard Sleijpen

Utrecht University
Mathematical Institute
room Freudenthal Building 504
Budapestlaan 6
Utrecht

Tel: +31-30-253 1732
Email: G.L.G.Sleijpen@uu.nl
http://www.staff.science.uu.nl/∼sleij101

‘Google’ for ‘Sleijpen’

Dr. ir. Martin van Gijzen

Delft University of Technology
Faculty EWI
room HB 07.260
Mekelweg 4
2628 CD Delft

Tel: +31 15-2782519
E-mail: M.B.vanGijzen@TUDelft.nl
http://ta.twi.tudelft.nl/nw/users/gijzen

Program Lecture 1

- Overview of the course
- Useful references
- A motivating example
- Review of basic linear algebra concepts
- Inner products, vector norms and matrix norms
- Condition number, finite precision arithmetic

Goals of the course

To provide **theoretical insight** and to develop **practical skills** for solving **numerically** large scale linear algebra problems.
Particular emphasis lies on large-scale linear systems and on eigenvalue problems.
At the end of the course you will

- understand the principles behind modern solution techniques for linear algebra problems;
- be able to implement them and to understand their behaviour;
- and you will be able to select (and adapt) a suitable method for your problem.

Topics per day (i.e., per lecture)

- Day 1: introduction, review of linear algebra, basics
- Day 2 and 3: direct solution methods
- Day 4: basic iterative methods for linear systems and eigenvalue problems
- Day 5-10: Krylov methods
- Day 10: preconditioning, parallel implementation
- Day 11: special topics
- Day 12 and 13: Eigenvalue problems
- Day 14: Subjects from Scientific Computing

Examination

- On Day 3: mandatory linear algebra review test. (Grade Q)
- Every week: homework assignments, to be handed in.
  The homework assignments must be made **individually**.

  (Grade H is the average of the 10 best)
- At the end of the lectures: final project assignment.
  You are allowed to do the project assignment in pairs. The report has to be handed in on January 31, 2018 at the latest. After handing it in you have to make an appointment to "defend" your report. (Grade P).

Final grade $C = 0.6 * P + 0.4 * H$ provided $Q \geq 6$, $H \geq 6$, $P \geq 5$.

Recommended literature

- Gene H. Golub and Charles F. Van Loan, **Matrix Computations**, The Johns Hopkins University Press, Baltimore.
- Henk van der Vorst, **Iterative methods for large linear systems**, Cambridge press, 2003.
- Richard Barrett, et al., **Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods**, SIAM.
- Zhaojun Bai et al., **Templates for the Solution of Algebraic Eigenvalue Problems**, SIAM.

Useful webpages

- http://www.netlib.org/: a wealth of information to numerical software and other information, e.g.
  - LAPACK, BLAS: dense linear algebra
  - NETSOLVE: grid computing
  - TEMPLATES: the book plus software
  - MATRIX MARKET: matrices
- http://www.math.uu.nl/people/vorst/: manuscript of the book, software
- http://www-math.mit.edu/ gs/: Gilbert Strang’s homepage: video course, demos ...

Motivation of the course

Many applications give rise to large linear algebra problems. Typically these problems involve matrices that are

- **Large**, 10^8 unknowns are not exceptional anymore;
- **Sparse**, only a fraction of the entries of the matrix is nonzero;
- **Structured**, the matrix often has a symmetric pattern and is banded.

Moreover, the matrix can have **special mathematical properties**, e.g., it may be symmetric, Toeplitz, or the eigenvalues may all be in the right-half plane.

An application: ocean circulation (1)

Physical model: balance between

- Wind force
- Coriolis force
- Bottom friction.

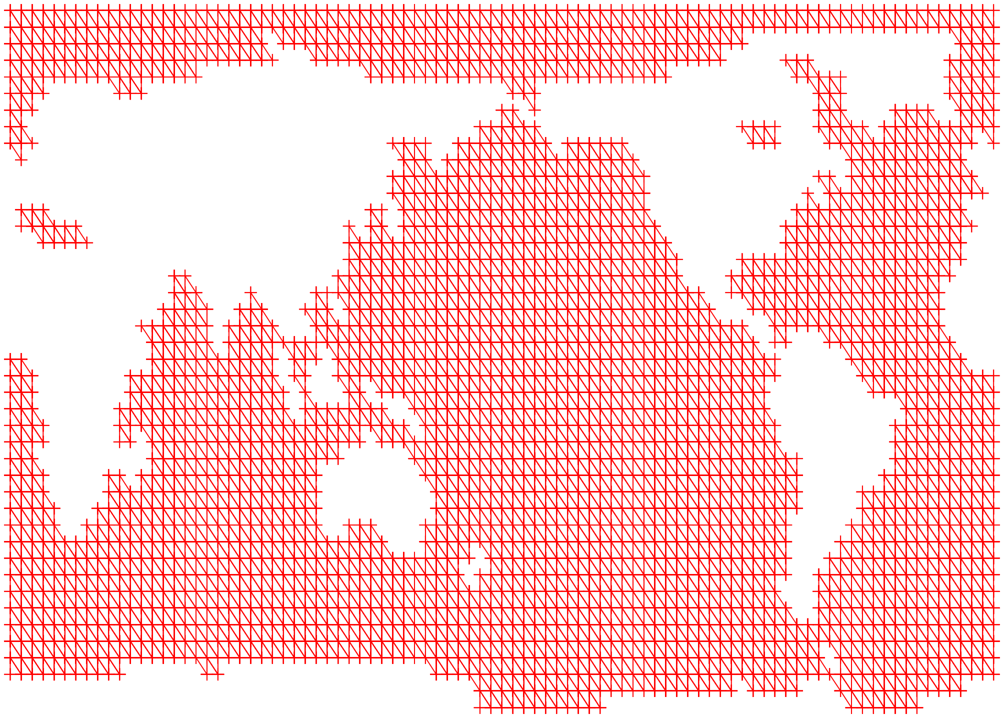
An application: ocean circulation (2)

Mathematical model

$$-r\nabla^2\psi - \beta\frac{\partial\psi}{\partial x} = \nabla \times \mathbf{F} \quad \text{on} \quad \Omega,$$

- ψ: stream function (to be computed)
- r: bottom friction parameter (available)
- β: Coriolis parameter
- $\mathbf{F}$: Wind stress (available from measurements)

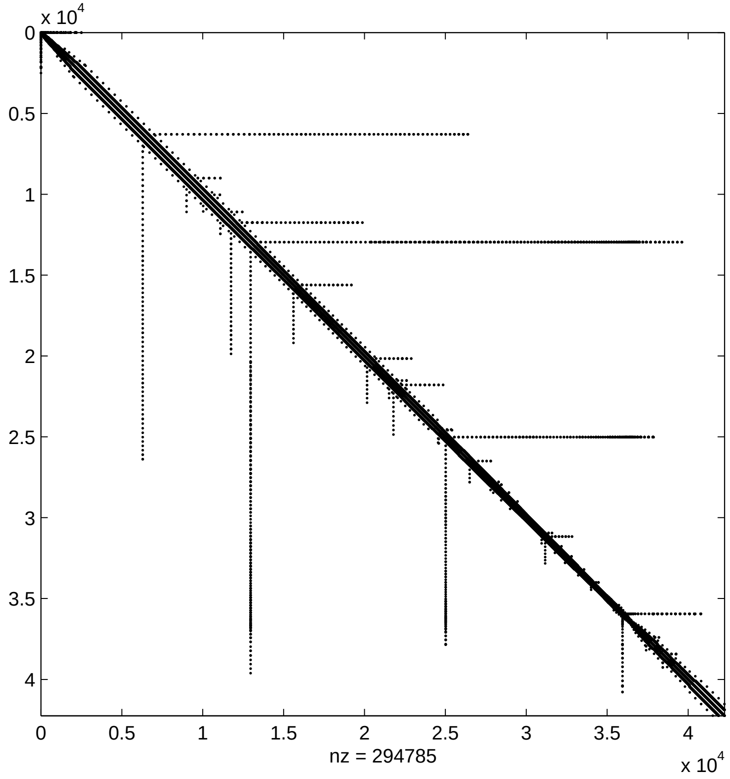
An application: ocean circulation (3)



Numerical model: discretization with FEM

An application: ocean circulation (4)

Discretization leads to a linear system $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{x}$ ($\sim \psi$) to be solved.



The nonzero pattern of the resulting matrix $\mathbf{A}$

Solving the resulting system

In order to be able to solve this problem you have to consider many questions:

- How can you exploit the sparsity of the matrix?
- Can you make use of the arrow structure?
- Is the matrix symmetric? Can you exploit this?
- Is the matrix close to singular? Is your solution algorithm sensitive to (numerical) errors?
- Can your solution method exploit the available (parallel) hardware?
- ...

Assignment 1a

The matrix

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

is the discretization of $-\frac{d^2y}{dx^2}$, with bc $\frac{dy}{dx}(0) = \frac{dy}{dx}(1) = 0$.

List as many characteristics of this matrix as possible.

Assignment 1a

The matrix

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

is the discretization of $-\frac{d^2y}{dx^2}$, with bc $\frac{dy}{dx}(0) = \frac{dy}{dx}(1) = 0$.

List as many characteristics of this matrix as possible.

- Symmetry? Positive definite?
- Rank? Range? Null space?
- Eigenvalues? Eigenvectors?

Assignment 1b

The matrix

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

is the (upwind) discretization of $\frac{dy}{dx}$, with bc $y(0) = 0$.

List as many characteristics of this matrix as possible.

Inner products

The **inner product** is a function $(\,.\,,\,.\,) : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ that satisfies the following three properties:

i) $(\mathbf{x}, \mathbf{y}) = \overline{(\mathbf{y}, \mathbf{x})}$ $\qquad (\mathbf{x}, \mathbf{y} \in \mathbb{C}^n)$,

ii) $(\mathbf{x} + \mathbf{y}, \mathbf{z}) = (\mathbf{x}, \mathbf{z}) + (\mathbf{y}, \mathbf{z})$, $\qquad (\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{C}^n)$,
and $(\alpha\mathbf{x}, \mathbf{y}) = \alpha(\mathbf{x}, \mathbf{y})$, $\qquad (\alpha \in \mathbb{C},\ \mathbf{x}, \mathbf{y} \in \mathbb{C}^n)$,

iii) $(\mathbf{x}, \mathbf{x}) \geq 0$, $\qquad (\mathbf{x}, \mathbf{x}) = 0 \;\Leftrightarrow\; \mathbf{x} = \mathbf{0}$, $\qquad (\mathbf{x} \in \mathbb{C}^n)$.

Example. $(\mathbf{x}, \mathbf{y}) = \mathbf{y}^*\mathbf{x}$ is the standard inner product.
$\mathbf{x}^* \equiv \mathbf{x}^H \equiv \overline{\mathbf{x}}^T$ denotes the conjugate transpose of $\mathbf{x}$.

Vector norms (1)

A **vector norm** on $\mathbb{C}^n$ is a function $\|\,.\,\| : \mathbb{C}^n \to \mathbb{R}$ that satisfies the following three properties:

i) $\|\mathbf{x}\| \geq 0 \qquad (\mathbf{x} \in \mathbb{C}^n),$

and $\|\mathbf{x}\| = 0 \;\Leftrightarrow\; \mathbf{x} = \mathbf{0}$,

ii) $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \qquad (\mathbf{x}, \mathbf{y} \in \mathbb{C}^n),$

iii) $\|\alpha\mathbf{x}\| = |\alpha|\,\|\mathbf{x}\| \qquad (\alpha \in \mathbb{C},\; \mathbf{x} \in \mathbb{C}^n)$.

Vector norms (2)

An important class of vector norms are the so-called **p-norms** (or Hölder norms), for $p \in [1, \infty]$, defined by

$$\|\mathbf{x}\|_p \equiv (|x_1|^p + \ldots + |x_n|^p)^{1/p}$$

for $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{C}^n$

The 1-,2-, and ∞-norms are the most commonly used

$$\|\mathbf{x}\|_1 = |x_1| + \ldots + |x_n|,$$

$$\|\mathbf{x}\|_2 = (|x_1|^2 + \ldots + |x_n|^2)^{1/2} = \sqrt{\mathbf{x}^*\mathbf{x}},$$

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

Assignment 2

Let the matrix $\mathbf{A}$ be Hermitian and Positive Definite.

- Does $(\mathbf{x}, \mathbf{y})_A \equiv \mathbf{x}^*\mathbf{A}\mathbf{y}$ define an inner product?
- Is the norm that is induced by this inner product a proper norm?

Orthogonality

Two vectors $\mathbf{x}$ and $\mathbf{y}$ are **orthogonal** with respect to an inner product if

$$(\mathbf{x}, \mathbf{y}) = 0.$$

The notation $\mathbf{x} \perp \mathbf{y}$ means that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal.
If the inner product is not specified the standard inner product $(p = 2)$ is assumed.

Two subspace $\mathcal{U}$ and $\mathcal{V}$ are **orthogonal** if for every $\mathbf{u} \in \mathcal{U}$ and $\mathbf{v} \in \mathcal{V}$ we have

$$(\mathbf{u}, \mathbf{v}) = 0.$$

Orthogonal matrices

A matrix (not necessarily square) is called **orthogonal** if

- All columns of the matrix are orthogonal (with respect to the standard inner product).

The matrix is **orthonormal** if, in addition,

- the columns are normalised.

Hence for an $n \times k$ orthonormal matrix $\mathbf{Q}$ we have $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, with $\mathbf{I}$ the $k \times k$ identity matrix.

Orthogonal matrices

A matrix (not necessarily square) is called **orthogonal** if

- All columns of the matrix are orthogonal (with respect to the standard inner product).

The matrix is **orthonormal** if, in addition,

- the columns are normalised.

Hence for an $n \times k$ orthonormal matrix $\mathbf{Q}$ we have $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, with $\mathbf{I}$ the $k \times k$ identity matrix.

Warning. Often a matrix $\mathbf{Q}$ is called orthogonal if it preserves orthogonality: $\mathbf{Qx} \perp \mathbf{Qy}$ whenever $\mathbf{x} \perp \mathbf{y}$. The columns of a matrix that preserves orthogonality have equal norm.

Orthogonal matrices

A matrix (not necessarily square) is called **orthogonal** if

- All columns of the matrix are orthogonal (with respect to the standard inner product).

The matrix is **orthonormal** if, in addition,

- the columns are normalised.

Hence for an $n \times k$ orthonormal matrix $\mathbf{Q}$ we have $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, with $\mathbf{I}$ the $k \times k$ identity matrix.

Orthogonal matrices allow stable computations.

Matrix norms (1)

The analysis of matrix algorithms frequently requires use of matrix norms.

For example, the quality of a linear system solver may be poor if the matrix of coefficients is "nearly singular".

To quantify the notion of near-singularity we need a measure of distance on the space $\mathbb{C}^{n\times k}$ of $n \times k$ matrices (with complex entries). Matrix norms provide that measure.

Matrix norms (2)

A **matrix norm** is a function $\|\,.\,\| : \mathbb{C}^{n\times k} \rightarrow \mathbb{R}$
that satisfies the following three properties:

i) $\|\mathbf{A}\| \geq 0 \qquad (\mathbf{A} \in \mathbb{C}^{n\times k}),$
and $\|\mathbf{A}\| = 0 \;\Leftrightarrow\; \mathbf{A} = \mathbf{0}$,

ii) $\|\mathbf{A}+\mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\| \qquad (\mathbf{A},\mathbf{B} \in \mathbb{C}^{n\times k}),$

iii) $\|\alpha\mathbf{A}\| = |\alpha|\,\|\mathbf{A}\| \qquad (\alpha \in \mathbb{C},\ \mathbf{A} \in \mathbb{C}^{n\times k})$.

The most commonly used matrix norms are the p-norms **induced** by the vector p-norms (for $p \in [1,\infty]$).

$$\|\mathbf{A}\|_p \equiv \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p=1} \|\mathbf{A}\mathbf{x}\|_p.$$

Matrix norms (3)

Below we list some properties of matrix p-norms

- $\|\mathbf{AB}\|_p \leq \|\mathbf{A}\|_p \, \|\mathbf{B}\|_p \qquad (\mathbf{A} \in \mathbb{C}^{n\times k},\ \mathbf{B} \in \mathbb{C}^{k\times m})$
- $\|\mathbf{A}\|_1 = \max\limits_{1\leq j\leq k} \sum\limits_{i=1}^{n} |a_{ij}| \qquad (\mathbf{A} = (a_{ij}) \in \mathbb{C}^{n\times k})$
- $\|\mathbf{A}\|_\infty = \max\limits_{1\leq i\leq n} \sum\limits_{j=1}^{k} |a_{ij}| \qquad (\mathbf{A} = (a_{ij}) \in \mathbb{C}^{n\times k})$
- $\|\mathbf{A}\|_2$ is equal to
  the square root of the largest eigenvalue of $\mathbf{A}^*\mathbf{A}$.
- All norms are **equivalent**, meaning that
  for all $p, q \in [1, \infty]$, $n, k \in \mathbb{N}$, there are $\kappa_1 \geq \kappa_0 > 0$
  such that $\kappa_0 \, \|\mathbf{A}\|_p \leq \|\mathbf{A}\|_q \leq \kappa_1 \, \|\mathbf{A}\|_p \qquad (\mathbf{A} \in \mathbb{C}^{n\times k})$.

Matrix norms (4)

Matrix norms that are not induced by a vector norm also exist. One of the best known is the **Frobenius norm**. The Frobenius norm of an $n \times k$ matrix $\mathbf{A} = (a_{ij})$ is given by

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{k} |a_{ij}|^2}$$

This is equal to

$$\|\mathbf{A}\|_F = \sqrt{\text{Trace}(\mathbf{A}^*\mathbf{A})}$$

In which Trace$(\mathbf{A})$ is the **trace** of $\mathbf{A}$, the absolute sum of the main diagonal elements of $\mathbf{A}$.

Condition number (1)

The condition number plays an important role in numerical linear algebra since it gives a measure on how perturbations in $\mathbf{A}$ and $\mathbf{b}$ affect the solution $\mathbf{x}$ of the system $\mathbf{Ax} = \mathbf{b}$.

The **condition number** $\mathcal{C}(\mathbf{A})$, for a non-singular matrix $\mathbf{A}$ (and matrix norm $\|\cdot\|$) is defined by

$$\mathcal{C}(\mathbf{A}) \equiv \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|.$$

We will see that a low condition number implies that small perturbations in the matrix or right-hand side give small changes in the solution.

A large condition number means that a small perturbation in the problem may give a large change in the solution.

Condition number (2)

Theorem. Suppose $\mathbf{A} \in \mathbb{C}^{n\times n}$ is non-singular, $\mathbf{b} \in \mathbb{C}^n$, $\mathbf{b} \neq \mathbf{0}$, and $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{A}(\mathbf{x} + \Delta_x) = \mathbf{b} + \Delta_b$, then

$$\frac{\|\Delta_x\|_p}{\|\mathbf{x}\|_p} \le \mathcal{C}_p(\mathbf{A}) \frac{\|\Delta_b\|_p}{\|\mathbf{b}\|_p} \ .$$

Here, $\mathcal{C}_p(\mathbf{A})$ is the condition number with respect to the p-norm.

Condition number (2)

Theorem. Suppose $\mathbf{A} \in \mathbb{C}^{n\times n}$ is non-singular, $\mathbf{b} \in \mathbb{C}^n$, $\mathbf{b} \neq \mathbf{0}$, and $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{A}(\mathbf{x} + \Delta_x) = \mathbf{b} + \Delta_b$, then

$$\frac{\|\Delta_x\|_p}{\|\mathbf{x}\|_p} \leq \mathcal{C}_p(\mathbf{A}) \, \frac{\|\Delta_b\|_p}{\|\mathbf{b}\|_p} \; .$$

Proof. From the properties of the norms it follows that

$$\|\mathbf{b}\|_p = \|\mathbf{A}\mathbf{x}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{x}\|_p, \quad \text{so} \quad \frac{1}{\|\mathbf{x}\|_p} \leq \|\mathbf{A}\|_p \frac{1}{\|\mathbf{b}\|_p}.$$

We know that $\mathbf{A}\Delta_x = \Delta_b$, so $\Delta_x = \mathbf{A}^{-1}\Delta_b$. Furthermore,

$$\|\Delta_x\|_p = \|\mathbf{A}^{-1}\Delta_b\|_p \leq \|\mathbf{A}^{-1}\|_p \|\Delta_b\|_p.$$

Combination of these inequalities proves the theorem.

Condition number (3)

Suppose, for $\mathbf{A} \in \mathbb{C}^{n\times n}$ non-singular and $\mathbf{b} \in \mathbb{C}^n$, $\mathbf{b} \neq \mathbf{0}$, you want the solution $\mathbf{x}$ of $\mathbf{A}\mathbf{x} = \mathbf{b}$.

You actually solve a perturbed system

$$(\mathbf{A} + \Delta_A)(\mathbf{x} + \Delta_x) = \mathbf{b} + \Delta_b$$

with $\Delta_A \in \mathbb{C}^{n\times n}$, $\|\Delta_A\|_p \leq \delta_A \|\mathbf{A}\|_p$
and $\Delta_b \in \mathbb{C}^n$, $\|\Delta_b\|_p \leq \delta_b \|\mathbf{b}\|_p$.

When has this system a (unique) solution?

Condition number (3)

Suppose, for $\mathbf{A} \in \mathbb{C}^{n\times n}$ non-singular and $\mathbf{b} \in \mathbb{C}^n$, $\mathbf{b} \neq \mathbf{0}$, you want the solution $\mathbf{x}$ of $\mathbf{Ax} = \mathbf{b}$.

You actually solve a perturbed system

$$(\mathbf{A} + \Delta_A)(\mathbf{x} + \Delta_x) = \mathbf{b} + \Delta_b$$

with $\Delta_A \in \mathbb{C}^{n\times n}$, $\|\Delta_A\|_p \leq \delta_A \|\mathbf{A}\|_p$
and $\Delta_b \in \mathbb{C}^n$, $\|\Delta_b\|_p \leq \delta_b \|\mathbf{b}\|_p$.

Theorem. If $\mu \equiv \mathcal{C}_p(\mathbf{A})\delta_A < 1$, then

$\mathbf{A} + \Delta_A$ is non-singular and $\dfrac{\|\Delta_x\|_p}{\|\mathbf{x}\|_p} \leq \dfrac{\mathcal{C}_p(\mathbf{A})}{1-\mu}(\delta_A + \delta_b)$.

Proof. see Golub and Van Loan, p.83.

Forward and backward error analysis

For the analysis of algorithms a **backward error analysis** is commonly used. In this approach the computed solution is viewed as the exact solution of a perturbed problem, and the question is how big the perturbations are.

If, for example, an algorithm for solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ yields a ‘solution’ $\widetilde{\mathbf{x}}$, then a backward error analysis

finds a Δ_A and a Δ_b with $\|\Delta_A\|$ and $\|\Delta_b\|$ small(est)

such that $\widetilde{\mathbf{x}}$ is the solution of $(\mathbf{A} + \Delta_A)\widetilde{\mathbf{x}} = (\mathbf{b} + \Delta_b)$.

(Δ_A, Δ_b) is called the **backward error**.

Forward and backward error analysis

For the analysis of algorithms a **backward error analysis** is commonly used. In this approach the computed solution is viewed as the exact solution of a perturbed problem, and the question is how big the perturbations are.

If, for example, an algorithm for solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ yields a 'solution' $\widetilde{\mathbf{x}}$, then a backward error analysis

finds a Δ_A and a Δ_b with $\|\Delta_A\|$ and $\|\Delta_b\|$ small(est)

such that $\widetilde{\mathbf{x}}$ is the solution of $(\mathbf{A} + \Delta_A)\widetilde{\mathbf{x}} = (\mathbf{b} + \Delta_b)$.

(Δ_A, Δ_b) is called the **backward error**.

Example. With $\Delta_b \equiv \mathbf{r} \equiv \mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}}$, we have $\mathbf{A}\widetilde{\mathbf{x}} = \mathbf{b} - \Delta_b$.

$\mathbf{r}$ is the **residual**: it gives a backward error for linear systems.

Forward and backward error analysis

For the analysis of algorithms a **backward error analysis** is commonly used. In this approach the computed solution is viewed as the exact solution of a perturbed problem, and the question is how big the perturbations are.

If, for example, an algorithm for solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ yields a ‘solution’ $\widetilde{\mathbf{x}}$, then a backward error analysis

finds a Δ_A and a Δ_b with $\|\Delta_A\|$ and $\|\Delta_b\|$ small(est)

such that $\widetilde{\mathbf{x}}$ is the solution of $(\mathbf{A} + \Delta_A)\widetilde{\mathbf{x}} = (\mathbf{b} + \Delta_b)$.

(Δ_A, Δ_b) is called the **backward error**.

In the context of a backward error $\mathbf{x} - \widetilde{\mathbf{x}}$ is called the **forward error**. A **forward error analysis estimates** the size of the forward error.

Finite precision

For some $\beta, t \in \mathbb{N}$, $L, U \in \mathbb{Z}$, a computer stores real numbers as

$$f = \pm 0.d_1 d_2 ... d_t \cdot \beta^e$$

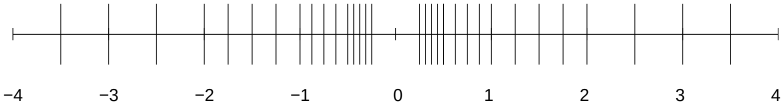
where $d_i \in \mathbb{Z},\ 0 \leq d_i < \beta,$ and $e \in \mathbb{Z},\ L \leq e \leq U$.

f is called a **floating point number**.

Let $\mathbb{F}$ denote the set of floating point numbers,

- β: **base**, ($\beta = 2$: binary, $\beta = 10$: decimal)
- t: **precision**, number of available digits
- $[L, U]$: **exponent range**

Note: floating point numbers are not equally spaced



Finite precision arithmetic

Each nonzero $f \in \mathbb{F}$ satisfies

$$m \leq |f| \leq M, \quad \text{where} \quad m \equiv \beta^{L-1} \quad \text{and} \quad M \equiv \beta^{U}(1-\beta^{-t}).$$

To have a model of computer arithmetic, the set $\mathbb{G}$ is defined by

$$\mathbb{G} \equiv \{x \in \mathbb{R} \mid m \leq |x| \leq M\} \cup \{0\},$$

and $\mathit{fl} : \mathbb{G} \to \mathbb{F}$ maps any real number x in $\mathbb{G}$ to a floating point number such that,

$$\mathit{fl}(x) = x(1+\xi) \quad \text{for some } \xi \text{ with } |\xi| \leq \mathrm{u}.$$

Here, u, is the **unit round-off** or **relative machine precision** defined by $\quad \mathrm{u} \equiv \frac{1}{2}\beta^{1-t} \qquad$ ($=$ `eps/2` $\approx 1.1\,10^{-16}$ in MATLAB).

Overflow and underflow

Let $\bullet$ represent a **flop** (*floating point operation*) or *basic arithmetic operation* (i.e., $+$, $-$, $*$, or $/$).

Let a and b be elements of $\mathbb{F}$.

If $|a \bullet b| \notin \mathbb{G}$, then an arithmetic fault occurs called

- **overflow** if $|a \bullet b| > M,$ and
- **underflow** if $0 < |a \bullet b| < m$.

Round-off in basic operations

Notation. $\mathit{fl}(a \bullet b)$ denotes the **computed version** of $a \bullet b$.

Computers are carefully constructed such that,
in case $a, b \in \mathbb{F}$,

$$\mathit{fl}(a \bullet b) = (a \bullet b)(1 + \xi) \quad \text{for some } \xi \text{ with } |\xi| \leq \mathsf{u}:$$

the computed quantity is the exact result with an **absolute relative error** at most u.

Flops with a zero $(a = 0$ or $b = 0)$ are exact.

For simplicity, we neglect the fact that flops as multiplication by 2 are exact as well.

Round-off in basic operations (2)

Example. Let $\alpha, x, y \in \mathbb{F}$. Then, for some $\xi_1, \xi_2 \in [-\mathbf{u}, \mathbf{u}]$

$$\widehat{z} \equiv \textit{fl}(\alpha x + y) = (\alpha x(1+\xi_1) + y)(1+\xi_2)$$

Hence, $\widehat{z} = z + \delta$ with

$$\begin{aligned} |\delta| &\leq \mathbf{u}(|\alpha|\,|x| + |\widehat{z}|) + \mathcal{O}(\mathbf{u}^2) \\ &\leq \mathbf{u}(2|\alpha|\,|x| + |y|) + \mathcal{O}(\mathbf{u}^2). \end{aligned}$$

Note.

- Here, (and in the future) we neglect $\mathcal{O}(\mathrm{u}^2)$ terms (for an exact bound, replace u by $\mathrm{u}/(1-\mathrm{u})$).
- $\alpha x + y$ is the typical scalar operation in NLA.

Round-off in basic operations (3)

Below $\mathbf{A}$ and $\mathbf{B}$ are matrices (of matching dimensions), $\mathbf{x}$ and $\mathbf{y}$ are vectors, α is a scalar.

Convention. For $\mathbf{A} = (a_{ij})$ and $\mathbf{B} = (b_{ij})$ in $\mathbb{C}^{m\times n}$,

- $|\mathbf{A}| \equiv (|a_{ij}|)$
- $\mathbf{B} \leq \mathbf{A}$ means $b_{ij}, a_{ij} \in \mathbb{R}$, $b_{ij} \leq a_{ij}$ (all i, j)

Round-off in basic operations (3)

Below $\mathbf{A}$ and $\mathbf{B}$ are matrices (of matching dimensions), $\mathbf{x}$ and $\mathbf{y}$ are vectors, α is a scalar.

Below all matrix -, vector - and scalar entries are in $\mathbb{F}$*.*

The following results are easy to show:

$$\mathit{fl}(\alpha\mathbf{A}) = \alpha\mathbf{A} + \mathbf{E} \text{ for some } \mathbf{E} \text{ such that } |\mathbf{E}| \le \mathrm{u}|\alpha\mathbf{A}|,$$

$$\mathit{fl}(\mathbf{A}+\mathbf{B}) = (\mathbf{A}+\mathbf{B}) + \mathbf{E} \text{ for some } \mathbf{E} \text{ s.t. } |\mathbf{E}| \le \mathrm{u}|\mathbf{A}+\mathbf{B}|.$$

Note that, in general, $\|\,|\mathbf{A}|\,\| \ne \|\mathbf{A}\|$.

$\|\,|\mathbf{A}|\,\|_p = \|\mathbf{A}\|_p$ for $p = 1, \infty, F$,

$\|\mathbf{A}\|_2 \le \|\,|\mathbf{A}|\,\|_2 \le \sqrt{n_A}\,\|\mathbf{A}\|_2$ with n_A max. number non-zeros per row of A.

Round-off in basic operations (4)

For the standard inner product (called DOT), we get

$$\mathit{fl}((\mathbf{x},\mathbf{y})_2) \equiv \mathit{fl}(\mathbf{y}^*\mathbf{x}) = (\mathbf{x},\mathbf{y})_2 + \delta, \quad \text{where}$$

$$\delta = (\Delta_x, \mathbf{y})_2 \;\text{ for some } \Delta_x \text{ s.t. } \; |\Delta_x| \leq n\,\mathbf{u}\,|\mathbf{x}| + \mathcal{O}(\mathbf{u}^2).$$

Round-off in basic operations (4)

For the standard inner product (called DOT), we get

$$\mathit{fl}((\mathbf{x},\mathbf{y})_2) \equiv \mathit{fl}(\mathbf{y}^*\mathbf{x}) = (\mathbf{x},\mathbf{y})_2 + \delta, \quad \text{where}$$

$$\delta = (\Delta_x, \mathbf{y})_2 \quad \text{for some } \Delta_x \text{ s.t. } \quad |\Delta_x| \leq n\,\mathbf{u}\,|\mathbf{x}| + \mathcal{O}(\mathbf{u}^2).$$

Notes.

- n can be replaced by $p_x \equiv \#\{j \mid x_j \neq 0\}$ (or by p_y).
- Here, (and in the future) we neglected $\mathcal{O}(\mathrm{u}^2)$ terms

(for an exact bound, typically replace $p_x\,\mathrm{u}$ by $p_x\,\mathrm{u}/(1 - p_x\,\mathrm{u})$).

- This leads to the bound $|\delta| \leq p_x\,\mathbf{u}\,(|\mathbf{x}|, |\mathbf{y}|)_2.$
- The error can also be represented by a perturbation on $\mathbf{y}$.

Round-off in basic operations (4)

For the standard inner product (called DOT), we get

$$\mathit{fl}((\mathbf{x},\mathbf{y})_2) \equiv \mathit{fl}(\mathbf{y}^*\mathbf{x}) = (\mathbf{x},\mathbf{y})_2 + \delta, \quad \text{where}$$

$$\delta = (\Delta_x, \mathbf{y})_2 \;\text{ for some } \Delta_x \text{ s.t. } \; |\Delta_x| \leq n\,\mathbf{u}\,|\mathbf{x}| + \mathcal{O}(\mathbf{u}^2).$$

If a number is the result of p_x basic arithmetic operations, then typically p_x shows up in the round-off error.

Round-off in basic operations (5)

For the scaled vector update (called AXPY) we get

$$\widehat{\mathbf{z}} \equiv \mathit{fl}(\alpha\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \mathbf{y} + \mathbf{e},$$

for some $\mathbf{e}$ such that $|\mathbf{e}| \leq \mathsf{u}\,(|\alpha\mathbf{x}| + |\widehat{\mathbf{z}}|)$

Note that the error involves the result $\widehat{\mathbf{z}}$ rather than $\mathbf{y}$:
the estimate $|\mathbf{e}| \leq \mathsf{u}\,(2\,|\alpha\mathbf{x}| + |\mathbf{y}|)$ is also correct but can be much bigger ($3\times$ if $\mathbf{y} \approx -\alpha\mathbf{x}$).

Round-off in basic operations (6)

If $\mathbf{A} \in \mathbb{C}^{n\times k}$, $\mathbf{B} \in \mathbb{C}^{k\times m}$, and p_A is the maximum of non-zeros per row of $\mathbf{A}$ (note that $p_A \leq n$), then

$$\mathit{fl}(\mathbf{AB}) = \mathbf{AB} + \mathbf{E} \quad \text{for some } \mathbf{E} \text{ s.t. } |\mathbf{E}| \leq p_A \, \mathsf{u} \, |\mathbf{A}| \, |\mathbf{B}|.$$

Notes.

- If $m = 1$, i.e., $\mathbf{B}$ is a vector, then, as for DOTs, $\mathbf{E}$ can be represented by a perturbation on $\mathbf{A}$:

$$\mathbf{E} = \Delta_A \mathbf{B} \quad \text{for some } \Delta_A \text{ s.t. } |\Delta_A| \leq p_A \, \mathbf{u} \, |\mathbf{A}|.$$

- There is *not* such an $\mathbf{A}$ perturbed represention if $m > 1$.

Similar results exist for all basic matrix and vector operations.

Full numerical accuracy

To verify that a vector $\mathbf{x}$ solves a problem, the only option usually is to check whether the residual is zero. Unfortunately, even if $\mathbf{x}$ is the exact solution, the *computed* residual will not be zero.

For instance, if $\mathbf{x}$ is the exact solution of the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A}$ a non-singular. Then , for the computed residual $\widehat{\mathbf{r}} \equiv \mathit{fl}(\mathbf{b} - \mathbf{A}\mathbf{x})$ we have the sharp bounds

$$|\widehat{\mathbf{r}}| \leq p_A\,\mathsf{u}\,|\mathbf{A}|\,|\mathbf{x}|, \qquad \frac{\|\widehat{\mathbf{r}}\|}{\|\mathbf{A}\|\,\|\mathbf{x}\|} \leq p_A\,\mathsf{u}\,\frac{\|\,|\mathbf{A}|\,\|}{\|\mathbf{A}\|}.$$

This implies that we can not distinguish an approximate solution $\widetilde{\mathbf{x}}$ from the true one if $|\mathit{fl}(\mathbf{b} - \mathbf{A}\widetilde{\mathbf{x}})| \leq p_A\,\mathsf{u}\,|\mathbf{A}|\,|\widetilde{\mathbf{x}}|$:
then $\widetilde{\mathbf{x}}$ has **full numerical accuracy** (is '**numerically exact**').

Concluding remarks

Today we saw some of the concepts that will allow us to answer questions like:

- How sensitive is my problem to perturbation?
- What is the effect of finite precision arithmetic? How sensitive is my algorithm for finite precision calculations?
- How accurate is my solution?

In the following lessons these and similar questions will play a crucial role.

Further reading: Golub and van Loan, Page 48 - 68