Modélisation et résolutions numérique et symbolique

via les logiciels Maple et Matlab

Jeremy Berthomieu Mohab Safey El Din Stef Graillat

Mohab.Safey@lip6.fr

Outline

- Previous course: partial review of what you should know;
- Linear Algebra: basic concepts;
- Linear Algebra: basic algorithms;
- Linear Algebra: Solving and CRT
- Linear Algebra: how to survive in Maple

Operations on integers

Assuming $\tau = \text{size}(n) \geq \text{size}(m)$,

Addition of n, m in $O(\text{size}(n))$ and size of the output is $\simeq \max(\text{size}(n), \text{size}(m))$

Multiplication of n, m:

$$\mathsf{M}_{\text{int}}(\tau) \in O(\tau^2) \qquad \text{(naive)}$$

$$\mathsf{M}_{\text{int}}(\tau) \in O(\tau^{1.59}) \qquad \text{(Karatsuba)}$$

$$\mathsf{M}_{\text{int}}(\tau) \in O(\tau \log \tau \log\log \tau) \qquad \text{(Fast Fourier Transform)}$$

and size of the output is $\simeq \text{size}(n) + \text{size}(m)$

GCD and Euclide's algorithm

Input: A, B integers
Output: R and U, V such that $R = \mathrm{GCD}(A, B)$ and $R = AU + BV$
Q, RS, US, VS are local variables which will store intermediate datas
Equations $R = AU + BV$ and $R' = AU' + BV'$ are loop invariants

```
#Initialize :
R := A, R' := B, U := 1, V := 0, U' := 0, V' := 1
while (R' <> 0) do
    Q := iquo(R, R');
    RS := R, US := U, VS := V;
    R := R', U := U', V := V';
    R' := RS - Q *R', U' = US - Q*U', V' = VS - Q*V';
od;
return R, U, V;
```

Complexity is in $O(\mathrm{size}(A)^2)$ (assuming $A \geq B$) – Proof is skipped.

Chinese Remainder Theorem

Let $P_1, \ldots, P_k$ be pairwise coprimes and $A_1, \ldots, A_k$ be a sequence of integers such that $A_i \in \frac{\mathbb{Z}}{P_i\mathbb{Z}}$.

There exists an integer X such that

$$\begin{aligned} X &= A_1 \mod P_1 \\ &\vdots \\ X &= A_k \mod P_k \end{aligned}$$

Moreover, all solutions are congruent modulo $N = P_1 \cdots P_k$.

Application: Choose $P_1, \ldots, P_k$ prime of size smaller than the word machine.

Chinese Remainder Reconstruction

Take $P = P_1 \times \cdots \times P_k$.
For $1 \leq i \leq k$,

- remark that $\frac{P}{P_i}$ and P_i are co-prime.
- take U_i as the inverse of $\frac{P}{P_i}$ modulo P_i

$$U_i \frac{P}{P_i} + V P_i = 1$$

Construct

$$X = A_1 \frac{P}{P_1} U_1 + \cdots + A_k \frac{P}{P_k} U_k$$

Examples

- Solve $X = 2 \mod 3$ and $X = 3 \mod 5$
- Solve $X = 1 \mod 2$ and $X = 2 \mod 3$ and $X = 3 \mod 5$

Basic algorithms in Linear Algebra

One of the most fundamental computational topics. Applications in:

- Graphs (think about adjacency matrices)
- Coding and Error-correcting codes (linear codes)
- Artifical Intelligence (discrimination, SVD, learning algorithms, etc.)
- Vision and Computer Graphics
- etc. $\rightsquigarrow$ it is everywhere.

Bibliography:

- *Modern Computer Algebra*, von zur Gathen, Gerhardt, Springer.
- *Méthodes matricielles – Introduction à la complexité algébrique*, Abdeljaoued, Lombardi, Springer.
- **Donald E. Knuth, Seminumerical algorithms, The Art of Computer Programming.**

We will see very elementary notions and algos with a focus on bit complexity and some tricks for implementations.

Vector spaces and linear applications

Let $\mathbb{K}$ be a field. We say that E is a $\mathbb{K}$-vector space if and only if:

- there exists $\mathbf{0} \in \mathbf{E}$ such that for all $u \in E$, $\mathbf{0} + u = u$
- for all $u \in E$, there exists v such that $u + v = 0$.
- for all u, v in E, $u + v \in E$;
- for all $\lambda \in \mathbb{K}$ and $v \in E$, $\lambda v \in E$.
- for $1 \in \mathbb{K}$ and all $u \in E$, $1.u = u$.
- addition (between elements of E) is associative and commutative.

Some examples:

- $\mathbb{K}^n$ is a vector space;
- $\mathbb{K}[X]$ is a $\mathbb{K}$-vector space.

Linear applications and finite dimensional vector spaces

We will be interested in vector spaces in which we can compute !
Basic operations are linear operations between vectors.

Linear maps

Let $\mathbb{K}$ be a field, E and F be two $\mathbb{K}$-vector spaces. We say that a map $\varphi : E \to F$ is linear if and only if:

- for all $\lambda \in \mathbb{K}$ and $u \in E$, $\varphi(\lambda u) = \lambda \varphi(u)$;
- for all u, v in E, $\varphi(u + v) = \varphi(u) + \varphi(v)$.

Basis

We say that $\mathscr{B} \subset E$ is a basis of E if and only if it is a set of linearly independent elements of E for all $u \in E$, there exists $b_1, \ldots, b_k$ in $\mathscr{B}$ and $\lambda_1, \ldots, \lambda_k$ in E such that $u = \lambda_1 b_1 + \cdots + \lambda_k b_k$.

Example: $1, X, X^2, \ldots, X^k, \ldots$ is a basis of $\mathbb{K}[X]$.

Matrices and linear applications

Let $\mathbb{K}$ be a field, E and F be $\mathbb{K}$-vector spaces and n, m be in $\mathbb{N}$. Let $\varphi : E \to F$ be a linear map.

Let $\mathscr{B}_E, \mathscr{B}_F$ be bases of E and F.

Finite dimensional vector spaces

We say that E has dimension n if and only if there exists a basis of $\mathscr{B}_E$ of E of cardinality n.

These are the vector spaces in which one can compute $\rightsquigarrow$ **why?**

Linear maps in finite dimensional vector spaces

The map φ is completly determined by the knowledge of $\varphi(\mathscr{B}_E)$ expressed in $\mathscr{B}_F$.

Matrices and linear applications

Let $\mathscr{B}_E = (\mathbf{e}_1, \ldots, \mathbf{e}_n)$ (resp. $\mathscr{B}_F = (\mathbf{f}_1, \ldots, \mathbf{f}_m)$) be a basis of E (F).

$$\begin{aligned} \varphi(\mathbf{e}_1) &= a_{1,1}\mathbf{f}_1 + \cdots + a_{m,1}\mathbf{f}_m \\ \vdots & \quad \vdots \\ \varphi(\mathbf{e}_n) &= a_{1,n}\mathbf{f}_1 + \cdots + a_{m,n}\mathbf{f}_m \end{aligned}$$

The $m \times n$ matrix $\Phi = \begin{bmatrix} a_{1,1} & \cdots & \cdots & a_{1,n} \\ \vdots & & & \vdots \\ a_{m,1} & \cdots & \cdots & a_{m,n} \end{bmatrix}$ is associated to φ.

Taking $\mathbf{a} = \lambda_1\mathbf{e}_1 + \cdots + \lambda_n\mathbf{e}_n$, $\varphi(\mathbf{a})$ is given by

$$\begin{bmatrix} a_{1,1} & \cdots & \cdots & a_{1,n} \\ \vdots & & & \vdots \\ a_{m,1} & \cdots & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \vdots \\ \lambda_n \end{bmatrix}$$

Data-structures and matrices

Data-structure: Basically, matrices are arrays on which we perform computations. Many data-structures can be used.

- Bi-dimensional arrays;
- uni-dimensional arrays (storing lines after lines or columns after columns the entries of the matrix)
- dedicated structures for **sparse** matrices

  $\rightsquigarrow$ less memory usage
- dedicated structures for **structured** matrices

  $\rightsquigarrow$ less memory usage

Basic operations on matrices (should be known)

- **Addition** of matrices (of the same sizes) is the addition of maps ($\theta = (\varphi + \psi)$ being defined by $\theta(u) = \varphi(u) + \psi(u)$) $\rightsquigarrow O(nm)$
- **Scalar-multiplication** of a matrix (by $\lambda \in \mathbb{K}$) is the multiplication (by λ of a map) $\rightsquigarrow O(nm)$

The set of $n \times m$ matrices is a $\mathbb{K}$-vector space.

- Multiplication matrix-vector provides the image of a vector by a map $\rightsquigarrow O(nm)$
- Multiplication matrix-matrix (with compatible sizes $n \times m$ and $m \times k$) is the composition of linear maps. $\rightsquigarrow O(nmk)$

Exercises: review algorithms and their complexities

Kernel and Images

Let $\varphi : E \to F$ be a linear map.

Kernel

$\ker(\varphi) = \{u \in E \mid \varphi(u) = 0\}$; it is a $\mathbb{K}$-vector subspace

Image

$\mathrm{Im}(\varphi) = \{v \in F \mid \exists u \in E,\ \varphi(u) = v\}$; it is a $\mathbb{K}$-vector subspace

Theorem (dimension formula)

$\dim(\mathrm{Im}(\varphi)) + \dim(\ker(\varphi)) = n$

- If $\ker(\varphi)$ is "big", $\mathrm{Im}(\varphi)$ is "small" and reciprocally.
- If $\ker(\varphi) = \{\mathbf{0}\}$ then φ is injective; If $n = m$ and $\ker(\varphi) = \{\mathbf{0}\}$ (square matrix), φ is an isomorphism.

The rank of a matrix

The rank of a matrix A is $\dim(\mathrm{Im}(A))$.

- it measures how close $\mathrm{Im}(A)$ is to the target space.
- it's also the number of linearly independent rows (resp. columns) of A.

Application

Let A be a $n \times n$ matrix and $\mathbf{v}$ be a vector. The system

$$A.X = \mathbf{v}$$

has a (unique) solution if $\ker(A) = \mathbf{0}$, i.e. $\mathrm{rank}(A) = n$!

We need a measure of rank defects $\rightsquigarrow$ determinants
This is relevant to computationally solve linear systems.

Determinant

Now we consider square matrices with entries in a field $\mathbb{K}$.

Formal definition

Let $A = (a_{i,j})_{1 \leq i,j \leq n}$ be a $n \times n$ matrix. The determinant of A is

$$\det(A) = \sum_{\sigma \in S_n} (-1)^{\mathrm{sign}(\sigma)} a_{1,\sigma(1)} \cdots a_{n,\sigma(n)}$$

where $\mathrm{sign}(\sigma) = \sharp\{1 \leq i < j \leq n \mid \sigma(i) > \sigma(j)\}$

Geometric interpretation and properties

- Determinants can be obtained by ring operations.
- Determinants and products of matrices: $\det(AB) = \det(A)\det(B)$
- Laplace expansion: $\det(A) = \sum_{1 \leq j \leq n} (-1)^{i+j} a_j A_{i,j}$ where $A_{i,j}$ is the submatrix obtained by deleting the row i and column j.
- Geometric interpretations:

$$\det(A) = 0 \Leftrightarrow \text{rows/columns are co-linear}$$

Determinants and solving linear systems

Now, we want to solve systems like $A.X = B$ where A is a $n \times n$ matrix and B is a $n \times 1$ vector

- This linear system has a unique solution if $\det(A) \neq 0$.
- One can use Cramer's formula

$$x_i = \frac{\det B_i}{\det A}$$

avec $B_i = \begin{pmatrix} a_{0,0} & \cdot & a_{0,i-1} & b_0 & a_{0,i+1} & \cdot & a_{0,n-1} \\ a_{1,0} & \cdot & a_{1,i-1} & b_1 & a_{1,i+1} & \cdot & a_{1,n-1} \\ \vdots & \vdots & \cdot & \vdots & & & \\ a_{n-1,0} & \cdot & a_{n-1,i-1} & b_{n-1} & a_{n-1,i+1} & \cdot & a_{n-1,n-1} \end{pmatrix}$

- Without any "smart" algorithm for computing determinants, one uses the formula based on cofactors $\rightsquigarrow O(n!)$
  Such a complexity is prohibitive

Hadamard's inequality

Let $A = (a_{i,j})_{1 \leq i,j \leq n}$ be a $n \times n$ matrix with entries in $\mathbb{Z}$.
Then

$$|\det(A)| \leq \prod_{j=1}^{n} \sqrt{\sum_{1 \leq i \leq n} a_{i,j}^2}$$

- Use in the scope of Chinese Remainder Theorem? $\rightarrow$ we need to lift elements of $\mathbb{Z}$!
- **Exercise:** Provide an upper bound on the size of $\det(A)$.

Eigenvalues – Eigenvectors

Definitions

We say that λ is an eigenvalue of A if and only if there exists $v \neq \mathbf{0}$ such that $Av = \lambda v$; in this case v is an eigenvector. The eigenspace E_λ associated to λ is the set of vectors v such that $Av = \lambda v$.

- E_λ is a vector subspace.
- λ is an eigenvalue of A if and only if it is a root of $\det(A - L\mathbf{Id})$.
- The ground field plays a particular role: if A is with entries in $\mathbb{K}$, eigenvalues may not lie in $\mathbb{K}$ but in a "larger" field (an algebraic closure).

Eigenvalues play a crucial role in many areas of computer science: data-mining (PageRank) – decision theory, imagery, scient. computing, etc.
BUT first we need good algorithms for solving linear systems (with polynomial bit complexity) $\rightarrow$ Determinants