

LECTURE 3: PROBLEMS AND ALGORITHMS

1. Mathematical problems

1.1. Formalism for defining a mathematical problem

In general, mathematical problems can be thought of as mappings from some set of inputs X to some set of outputs Y . The mapping is often carried out through a function f , i.e., a procedure that associates a single $y \in Y$ to some input $x \in X$

$$f: X \rightarrow Y, y = f(x), x \xrightarrow{f} y$$

Examples:

- Compute the square of a real:

$$X = \mathbb{R}, Y = \mathbb{R}, y = f(x) = x^2.$$

- Find x solution of $ax + b = c$ for given $a, b, c \in \mathbb{R}$, $a \neq 0$. The inputs to this problem are a, b, c and the output is the solution $(c - b) / a$

$$X = \mathbb{R} \setminus \{0\} \times \mathbb{R} \times \mathbb{R}, Y = \mathbb{R}, f(a, b, c) = (c - b) / a.$$

- Compute the inner product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$:

$$X = \mathbb{R}^n \times \mathbb{R}^n, Y = \mathbb{R}, y = f(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^n u_i v_i$$

with u_i, v_i the components of \mathbf{u}, \mathbf{v} . Note that the input set is the Cartesian product of sets of vectors and the output set is the reals. Such functions defined from sets of vectors (more accurately vector spaces) to reals (more accurately scalars) are called *functionals*.

- Compute the definite integral

$$(u, v) = \int_a^b u(x) v(x) dx,$$

with u, v arbitrary continuous functions, denoted by $u, v \in C^{(0)}([a, b])$:

$$X = C^{(0)}([a, b]) \times C^{(0)}([a, b]), Y = \mathbb{R}.$$

Again, this an example of a functional.

- Compute the derivative of a function $g \in C^{(1)}(\mathbb{R})$, with $C^{(k)}(\mathbb{R})$ the space of functions defined on \mathbb{R} differentiable k times: $X = C^{(1)}(\mathbb{R})$, $Y = C^{(0)}(\mathbb{R})$, $f = d/dx$. Note that in this case X, Y are sets of functions, in which case f is referred to as an *operator*.
- Find the roots of a polynomial $p_n(x) = a_n x^n + \dots + a_1 x + a_0$. The input is the polynomial specified by the vector of coefficients $\mathbf{a} \in \mathbb{R}^{n+1}$. The output is another vector $\mathbf{x} \in \mathbb{R}^n$ whose components are roots, $p_n(x_i) = 0$

$$X = \mathbb{R}^{n+1}, Y = \mathbb{R}^n.$$

The function $f: X \rightarrow Y$ cannot be written explicitly (corollary of Abel-Ruffini theorem), but there are approximations \tilde{f} of the root-finding function that can be implemented such $\tilde{f} \cong f$.

Note that the specification of a mathematical problem requires definition of the triplet (X, Y, f) .

Once a problem is specified, the natural question is to ascertain whether a solution is possible. Generally, simple affirmation of the existence of a solution is the objective of some field of mathematics (e.g., analysis, functional analysis). From the point of view of science, an essential question is not only existence but also:

1. how does the output $y = f(x)$ change if x changes?
2. what are the constructive methods to approximate y ?

1.2. Vector space

The above general definition of a mathematical problem must be refined in order to assess magnitude of changes in inputs or outputs. A first step is to introduce some structure in the input and output sets X, Y . Using these sets, vector spaces $\mathcal{V} = (V, S, +, \cdot)$ are constructed, consisting of a set of vectors V , a set of scalars S , an addition operation $+$, and a scaling operation \cdot . The vector space is often referred to simply by its set of vectors V , when the set of scalars, addition operation, and scaling operation are self-evident in context.

Formally, a *vector space* \mathcal{V} is defined by a set V whose elements satisfy certain scaling and addition properties, denoted all together by the 4-tuple $\mathcal{V} = (V, S, +, \cdot)$. The first element the 4-tuple is a set whose elements are called *vectors*. The second element is a set of scalars, and the third is the vector addition operation. The last is the scaling operation, seen as multiplication of a vector by a scalar. The vector addition and scaling operations must satisfy rules suggested by positions or forces in three-dimensional space, which are listed in Table 1. In particular, a vector space requires definition of two distinguished elements: the zero vector $\mathbf{0} \in V$, and the identity scalar element $1 \in S$.

Addition rules for $\forall a, b, c \in V$	
$a + b \in V$	Closure
$a + (b + c) = (a + b) + c$	Associativity
$a + b = b + a$	Commutativity
$\mathbf{0} + a = a$	Zero vector
$a + (-a) = \mathbf{0}$	Additive inverse
Scaling rules for $\forall a, b \in V, \forall x, y \in S$	
$xa \in V$	Closure
$x(a + b) = xa + xb$	Distributivity
$(x + y)a = xa + ya$	Distributivity
$x(ya) = (xy)a$	Composition
$1a = a$	Scalar identity

Table 1. Vector space $\mathcal{V} = (V, S, +, \cdot)$ properties for arbitrary $a, b, c \in V$

1.3. Norm

A first step is quantification of the changes in input or output, assumed to have the structure of a vector space, $\mathcal{X} = (X, \mathbb{R}, +, \cdot)$, $\mathcal{Y} = (Y, \mathbb{R}, +, \cdot)$.

DEFINITION 1. A norm on vector space \mathcal{X} is a function $\|\cdot\|: X \rightarrow \mathbb{R}_+$, that for any $x, y, z \in X$, $a \in \mathbb{R}$ satisfies the properties:

1. $\|x\| = 0$ if and only if $x = \mathbf{0}$.
2. $\|ax\| = |a|\|x\|$
3. $\|x + y\| \leq \|x\| + \|y\|$

1.4. Condition number

The ratio of changes in output to changes in input is the absolute condition number of a problem.

DEFINITION 2. The problem $f: X \rightarrow Y$ has absolute condition number

$$\hat{\kappa} = \lim_{\varepsilon \rightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x + \delta x) - f(x)\|}{\|\delta x\|}$$

To avoid influence of choice of reference unit, the relative condition number is also introduced.

DEFINITION 3. The problem $f: X \rightarrow Y$ has relative condition number

$$\kappa = \lim_{\varepsilon \rightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x + \delta x) - f(x)\|}{\|f(x)\|} \cdot \frac{\|x\|}{\|\delta x\|}.$$

2. Solution algorithm

2.1. Accuracy

In scientific computation, the mathematical problem $f: X \rightarrow Y$ is approximated by an *algorithm* $\tilde{f}: \tilde{X} \rightarrow \tilde{Y}$, in which is assumed to be computable, and \tilde{X}, \tilde{Y} are vector spaces that approximate X, Y . As a first step in characterizing how well the algorithm \tilde{f} approximates the problem f , consider that $\tilde{X} = X$ and $\tilde{Y} = Y$, i.e., there is no error in representation of the domain and codomain.

DEFINITION 4. The absolute error of algorithm $\tilde{f}: X \rightarrow Y$ that approximates the problem $f: X \rightarrow Y$ is

$$e = \|\tilde{f}(x) - f(x)\|.$$

DEFINITION 5. The relative error of algorithm $\tilde{f}: X \rightarrow Y$ that approximates the problem $f: X \rightarrow Y$ is

$$\varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}.$$

DEFINITION 6. An algorithm $\tilde{f}: X \rightarrow Y$ is accurate if there exists finite $M \in \mathbb{R}_+$ such that

$$\varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq M \epsilon_{\text{mach}}$$

The above condition is also denoted as $\varepsilon = \mathcal{O}(\epsilon_{\text{mach}})$

2.2. Stability

Algorithms should not catastrophically increase input errors. This is quantified in the concept of stability.

DEFINITION 7. An algorithm $\tilde{f}: X \rightarrow Y$ is forward stable if

$$\|\tilde{x} - x\| / \|x\| = \mathcal{O}(\epsilon_{\text{mach}}) \Rightarrow \|\tilde{f}(x) - f(\tilde{x})\| / \|f(\tilde{x})\| = \mathcal{O}(\epsilon_{\text{mach}})$$

The above states that the relative error in the output should be on the order of machine epsilon if the relative in the input is of order machine epsilon. Note that the constants in the order statements M, N are usually different from one another, $\|\tilde{x} - x\| / \|x\| \leq M \epsilon_{\text{mach}}$, $\|\tilde{f}(x) - f(\tilde{x})\| / \|f(\tilde{x})\| \leq N \epsilon_{\text{mach}}$.

DEFINITION 8. An algorithm $\tilde{f}: X \rightarrow Y$ is backward stable if from existence of some \tilde{x} such that $\tilde{f}(x) = f(\tilde{x})$, it results that

$$\|\tilde{x} - x\| / \|x\| = \mathcal{O}(\epsilon_{\text{mach}}).$$

Backward stability asserts that the result of the algorithm on exact input data is the same as the solution to the mathematical problem for nearby data (with distance on order of machine epsilon).

Summary.

- Mathematical problems are stated as functions from a set of inputs X to a set of outputs Y , $f: X \rightarrow Y$
- The difficulty of a mathematical problem is assessed by measuring the effect of changes in input
- To quantify changes in inputs and outputs, the framework of a normed vector space is introduced
- The ratio of norm of output change to norm of input change is the absolute condition number of a problem

$$\hat{\kappa} = \lim_{\varepsilon \rightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x + \delta x) - f(x)\|}{\|\delta x\|}$$

- Algorithms are constructive approximations of mathematical problems $\tilde{f}: X \rightarrow Y$. The accuracy of an algorithm is assessed by comparison of the algorithm output to that of the mathematical problem through absolute error e and relative error ε

$$e = \|\tilde{f}(x) - f(x)\|, \varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

- The tendency of an algorithm to amplify perturbations of input is assessed by the concept of stability
- Algorithms that do not amplify relative changes in input of the size of machine precision are forward stable.
- Algorithms that compute the exact result of a mathematical problem for changes in put of the size of machine precision are backward stable.