Lecture 22 (hrs. 43,44) - November 13, 2025, 8:30 - 10:30 F3

(2.59) Example (machine numbers in *Scilab*).

In *Scilab* the set of machine numbers is:

$$M = F_d(2,53,-1021,1024)$$

that is, the set of floating point numbers, base two, precision 53, *bounded exponent* (between -1021 and 1024) and *with denormalized elements*.

The elements of M are:

- zero;
- the *normalized elements*:

$$(-1)^s\ 2^b\ 0.c_1...c_{53}$$

  where $s \in \{0,1\}$, $-1021 \leqslant b \leqslant 1024$, each c_k is a base two digit and $c_1 \neq 0$;
- the *denormalized elements*:

$$(-1)^s\ 2^{-1024}\ 0.c_1...c_{53}$$

  where $s \in \{0,1\}$, each c_k is a base two digit and $c_1 = 0$.

The set M has a *finite* number of elements. Furthermore:

- $\max M = \xi_{max} = 2^{1024}\, 0.1\cdots1 = 2^{1024}\,(1 - 2^{-53})$
- $\min\{\ \xi \in M,\ \xi > 0\ \} = \xi_{min} = 2^{-1021}\, 0.0\cdots01 = 2^{-1021}\, 2^{-53} = 2^{-1074}$
- the *successor of zero* is defined and: $\sigma(0) = \xi_{min}$
- $\min\{\ \xi \in M,\ \xi > 0$ and ξ is a normalized element $\} = 2^{-1021}\, 0.10\cdots0 = 2^{-1021}\, 2^{-1} = 2^{-1022}$
- M contains *symbolic elements*: Nan (used when the result of a predefined function cannot be assigned a ‘well defined’ numeric value), Inf (when a predefined function returns a ‘too large’ positive numeric value), -Inf (when a predefined function returns a ‘too large’ negative numeric value); in *Scilab* the constants %nan and %inf have values Nan and Inf respectively.
- let $rd: R \to M$ be the usual rounding function in M; the $rd^*: R \to M$ function that Scilab uses to round elements is defined as follows:

  if $|rd(x)| \leqslant \xi_{max}$ then $rd^*(x) = rd(x)$

  if $rd(x) > \xi_{max}$ then $rd^*(x) = Inf$

  if $rd(x) < -\xi_{max}$ then $rd^*(x) = -Inf$

The built-in *Scilab* function *number_properties* returns information about the set M. Specifically:

number_properties(<string>)

returns:

- the *base* of the set M when <string> = 'radix'
- the *precision* of the set M when <string> = 'digits'

- the *minimum exponent* of the set M when <string> = 'minexp'
- the *maximum exponent* of the set M when <string> = 'maxexp'
- the *presence of denormalized elements* when <string> = 'denorm'
- the *maximum element* of M when <string> = 'huge'
- the *minimum positive element* of M when <string> = 'tiniest'
- the *minimum normalized positive element* of M when <string> = 'tiny'
- the *machine precision* in M when <string> = 'eps'

The built-in *Scilab* function *log2* returns the fraction and exponent of an element of M. Specifically, if $\xi = (-1)^s\ 2^b\ g$, the assignment:

$$[f,e] = \log2(\xi)$$

sets $f = (-1)^s g$ and $e = b$.

The built-in *Scilab* function *nearfloat* returns the predecessor or the successor of an element of M. Specifically:

nearfloat(<string>,ξ)

returns:

- the *successor* of ξ when <string> = 'succ'
- the *predecessor* of ξ when <string> = 'pred'

(2.60) Homework.

Execute and discuss (using appropriate graphical representations) the following dialogues in *Scilab*:

```
> xi_min = number_properties('tiniest')
> xi_min == 2^(-1074)
> [f,e] = log2(xi_min)
> y = xi_min / 2
> y == 0
> z = 2^(-1075) * (3 / 2)
> z == 0
> z = xi_min * (3 / 4)
> z == xi_min
> xi_max = number_properties('huge')
> [f,e] = log2(xi_max)
> f == 1 - 2^(-53)
> xi_max + 2^971[1]
> nearfloat('succ',xi_max)
> xi_max + 2^970
> xi_max + 2^969 == xi_max
```

(2.61) Homework.

The built-in *Scilab* function *bitstring* returns the base-two string of digits that represents the usual encoding of a machine number in the computer. See the Wikipedia page: Double-precision floating-point format to ‘decipher’ the result of the following dialogue in *Scilab*:

---

1 The distance between xi_max and its successor in F(2,53) is $2^{1024-53} = 2^{971}$.

```
> bitstring(1)
> bitstring(xi_min)
> bitstring(0)
> bitstring(%inf)
```

(2.62) Remark.

Given $H \in R^{n \times n}$ such that $I - H$ is invertible, set:

$$C = \{ g \in R^n \text{ s.t. } x(k) \text{ is a convergent sequence}\}$$

one and only one of the following eventualities subsists:

(1) C has only *one element* (the solution of the system $(I - H)\, x = c$)

(2) C is a *vector subspace of* R^n *of dimension* $\leqslant n$ (determined by the eigenvectors of H)

(3) $C = R^n$

If one of the cases (1) or (2) holds, it is *practically impossible* to determine g such that the sequence x(k) is convergent: the method *cannot be used* to approximate the solution of $A\,x = b$.

If case (3) holds, any g generates a sequence convergent to the solution of the system $A\,x = b$: the method *can be used* to approximate the solution of $A\,x = b$.

(2.63) Definition (convergent method).

Let $H \in R^{n \times n}$ and $c \in R^n$. *The iterative method* defined by H and c *is convergent* if:

(1) for every $g \in R^n$, *the sequence* x(k) generated by the method starting from g *is convergent*;

(2) all sequences generated by the method have *the same limit*.

(2.64) Remark.

In the (usual) case where the iterative method is used to approximate the solution of the system $A\,x = b$ where A is an invertible matrix, the systems $A\,x = b$ and $(I - H)\,x = c$ are equivalent, and hence the method defined by H and c has *only one fixed point*. In this case (see Remark (2.57) of Lecture 21) we have that (1) $\Rightarrow$ (2), that is: the iterative method is convergent means that all the sequences generated by the method are convergent.

(2.65) Defifnition (spectrum and spectral radius).

Let $A \in R^{n \times n}$. The set of eigenvalues of A is called the *spectrum* of A:

$$\sigma(A) = \{ \lambda \in C \text{ s.t. } \lambda \text{ is an eigenvalue of } A \}$$

The *spectral radius* of A is the number:

$$\rho(A) = \max \{ |\lambda| \text{ s.t. } \lambda \text{ is an eigenvalue of } A \}$$[2]

(2.66) Theorem (characterization of convergent methods).

Let $H \in R^{n\times n}$ and $c \in R^n$. The iterative method defined by H and c is convergent *if and only if* $\rho(H) < 1$.

(2.67) Example.

(1) Let $H = \begin{bmatrix} 1/2 & 0 \\ 0 & -1 \end{bmatrix}$, c = 0 and $g \in R^2$. The sequence generated by the iterative method defined by H and c starting from g is:

$$x(k) = H^k g = \begin{bmatrix} (1/2)^k & 0 \\ 0 & (-1)^k \end{bmatrix} = \begin{bmatrix} (1/2)^k g_1 \\ (-1)^k g_2 \end{bmatrix}$$

The sequence is convergent (to the unique fixed point of the method: 0) if and only if $g_2 = 0$. Therefore the method is *not* convergent. In fact: $\sigma(H) = \{ 1/2, -1 \}$ and $\rho(H) = 1$.

(2) Let $H = \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix}$, c = 0 and $g \in R^2$. The sequence generated by the iterative method defined by H and c starting from g is:

$$x(k) = H^k g = \begin{bmatrix} (1/2)^k & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} (1/2)^k g_1 \\ g_2 \end{bmatrix}$$

The sequence is convergent for every g and:

$$\lim_{k\to\infty} \begin{bmatrix} (1/2)^k g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} 0 \\ g_2 \end{bmatrix}$$

The value of the limit *depends on* g, so the method is *not* convergent. In fact: $\sigma(H) = \{ 1/2, 1 \}$ and $\rho(H) = 1$.

---

2 Let us represent the eigenvalues of A, that is, $\sigma(A)$, on the complex plane. Given a sufficiently large positive real number r, the set $I(0,r) = \{ z \in C : | z | \leqslant r \}$ - the circle with center at the origin and radius r - contains $\sigma(A)$. The spectral radius of A is the *minimum* value of r such that $I(0,r) \supset \sigma(A)$.