Intervalle de représentation en précision finie

- On considère $\mathrm{F}(\beta,\mathrm{t}, e_{min}, e_{max})$: ensemble des nombres représentés via IEEE754
  - Donc $f \in F, \ |f| = 1,m \times 2^e$

0	e + biai (1023)	X	X	X	X	…	X	X	X

11 bits de l'exposant

52 bit de la mantisse
(1 sur le bit caché)

  - $0 \leq m \leq 2^t - 1$
    - $0 \leq (m \times 2^{-t}) \leq (2^t - 1).2^{-t}$

Intervalle de représentation en précision finie

- $0 \leq 0, m \ \leq 1 - 2^{-t}$
- $1 \leq 1 + (m \ \times 2^{-t}) \leq 2 - 2^{-t}$
- $1 \times \ 2^{e} \leq 1 + (m \ \times 2^{-t}) 2^{e} \leq (2 - 2^{-t}) \times \ 2^{e}$
- Mais $e \in [e_{min}, e_{max}]$
- $2^{e_{min}} \leq |f| \leq (2 - 2^{-t}) \times 2^{e_{max}}$

Intervalle de représentation en précision finie

- $2^{e_{min}} \leq |f| \leq (2 - 2^{-t}) \times 2^{e_{max}}$

Nombres standards à double précision F(2,52,-1022,1023)

$$2^{-1022} \leq |f| \leq (2 - 2^{-52}) \times 2^{1023}$$

$2{,}2250738585072020\ 10^{-308} \leq |f| \leq 1{,}7976931348623157\ 10^{308}$

Nombres standards à simple précision F(2,23,-126,127)

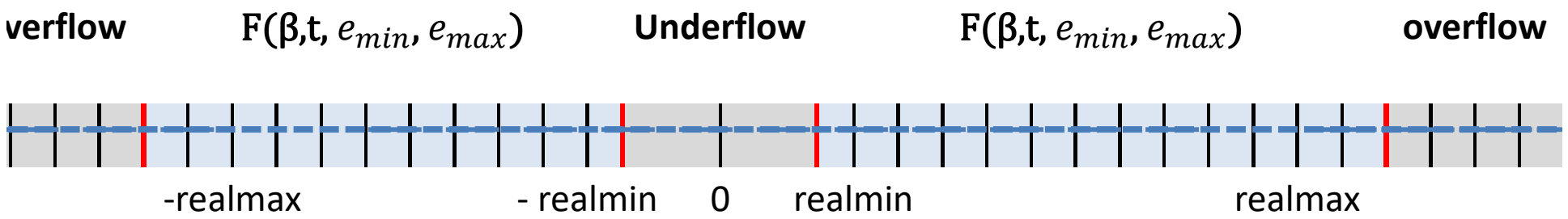
$$2^{-126} \leq |f| \leq (2 - 2^{-52}) \times 2^{127}$$

$1{,}175494351\ 10^{-38} \leq |f| \leq 3{,}4028235\ 10^{38}$

Débordement

Est-ce qu'on peut représenter une valeur plus importante ou encore plus petite?

- Débordement vers l'infini (overflow): dépassement de la capacité de stockage
- Débordement vers zéro (underflow): si le nombre est inférieur à la valeur positive minimale représentable.



Débordement

Solutions

❖Overflow :

S	E	F	Représentation binaire			Valeur spéciale
0	0	0	0	00000000000	00 ··· 00	0
1	0	0	1	00000000000	00 ··· 00	-0
0	2 047	0	0	11111111111	00 ··· 00	$+\infty$
1	2 047	0	1	11111111111	00 ··· 00	$-\infty$
0 ou 1	2 047	$F \neq 0$	S	11111111111	F	NaN[a]

[a] *Not a number*, par exemple $\frac{0}{0}$ ou $\frac{\infty}{\infty}$.

❖ underflow: un remplacement par 0 est en général effectué,

Quelle est la distance entre deux nombres consécutifs en précision finie ?

$$1 + \varepsilon_{machine} \neq 1$$

- Représentation du 1 (IEEE754) = +1,0 x 2^0

0	0 1 1 … 1 1	0 0 0 0 … 0 0 0

11 bits de l'exposant (0+1023)

52 bit de la mantisse (1 sur le bit caché)

- Représentation du suiv(1) (IEEE754)=+(1,0+2^{-52})x 2^0

0	0 1 1 … 1 1	0 0 0 0 … 0 0 1

11 bits de l'exposant (0+1023)

52 bit de la mantisse (1 sur le bit caché)

Quelle est la distance entre deux nombres consécutifs en précision finie ?

0	0	1	1	…	1	1	0	0	0	0	…	0	0	0

0	0	1	1	…	1	1	0	0	0	0	…	0	0	1

Pour une mantisse comportant t bits on a :

$$\text{eps}_{\text{machine}} = \varepsilon_M = 2^{-t}$$

= Distance qui sépare le nombre 1 du plus proche nombre flottant suivant.

eps(double précision) = $2^{-52} \approx 2.22 \times 10^{-16}$.

eps (simple précision)= $2^{-23} \approx 1.1921 \times 10^{-07}$

Quelle est la distance entre deux nombres consécutifs en précision finie ?

Exemple: Déterminer les points du système F(2,3,-2,2)

$f = n \times 2^{-t} \times 2^{e} = n \times 2^{e-t}$									
n					2^{e-t}				
					$e=-2$	$e=-1$	$e=0$	$e=1$	$e=2$
					1/32	1/16	1/8	1/4	1/2
1	0	0	0	=8	1/4	1/2	1	2	4
1	0	0	1	=9	9/32	9/16	9/8	9/4	9/2
1	0	1	0	=10	5/16	5/8	5/4	5/2	5
1	0	1	1	=11	11/32	11/16	11/8	11/4	11/2
1	1	0	0	=12	12/32	3/4	3/2	3	6
1	1	0	1	=13	13/32	13/16	13/8	13/4	13/2
1	1	1	0	=14	14/32	7/8	7/4	7/2	7
1	1	1	1	=15	15/32	15/16	15/8	15/4	15/2

Quelle est la distance entre deux nombres consécutifs en précision finie ?

Command Window

```
 Expression                          Return Value
=============================================
 eps(1/2)                            2^(-53)
 eps(1)                              2^(-52)
 eps(2)                              2^(-51)
 eps(realmax)                        2^971
 eps(0)                              2^(-1074)
 eps(realmin/2)                      2^(-1074)
 eps(realmin/16)                     2^(-1074)
 eps(Inf)                            NaN
 eps(NaN)                            NaN
---------------------------------------------
 eps(single(1/2))                    2^(-24)
 eps(single(1))                      2^(-23)
 eps(single(2))                      2^(-22)
 eps(realmax('single'))              2^104
 eps(single(0))                      2^(-149)
 eps(realmin('single')/2)           2^(-149)
 eps(realmin('single')/16)          2^(-149)
 eps(single(Inf))                    single(NaN)
 eps(single(NaN))                    single(NaN)
```

Quelle est la distance entre deux nombres consécutifs en précision finie ?

Les éléments de $\mathbb{F}$ sont "plus denses" quand on s'approche de x_{min}, et "moins denses" quand on s'approche de x_{max}. Ainsi, le nombre de $\mathbb{F}$ le plus proche de x_{max} (à sa gauche) et celui le plus proche de x_{min} (à sa droite), sont respectivement

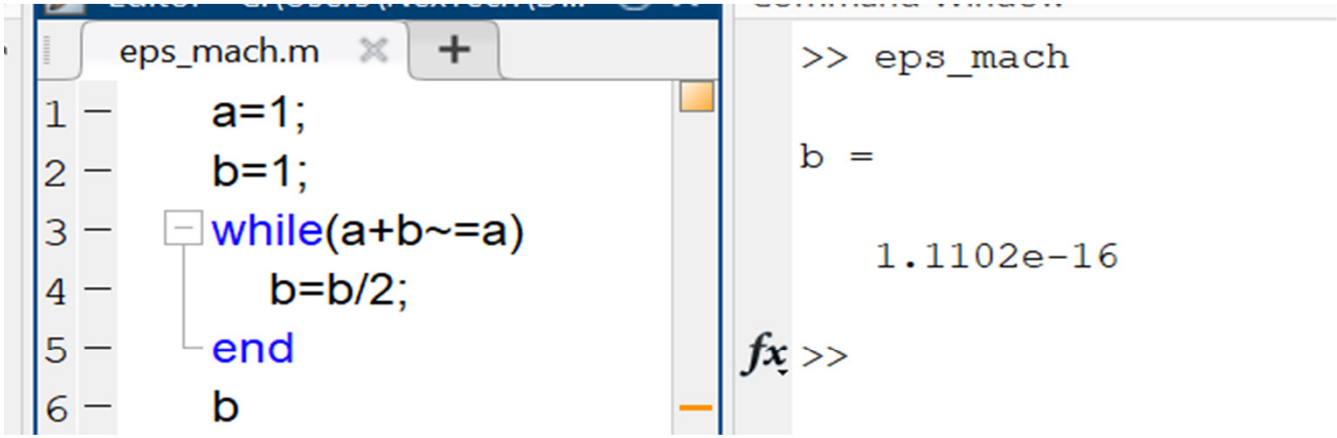
$$x^-_{max} = 1.797693134862315 \cdot 10^{+308},$$
$$x^+_{min} = 2.225073858507202 \cdot 10^{-308}.$$

On a donc $x^+_{min} - x_{min} \simeq 10^{-323}$, tandis que $x_{max} - x^-_{max} \simeq 10^{292}$ (!). Néanmoins, la distance relative est faible dans les deux cas,

- La distance df entre un nombre à virgule flottante normalisé non nul f et le nombre à VF normalisé adjacent est :

$$\beta^{-1} \ \varepsilon_{mach} \ |f| \leq df \leq \varepsilon_{mach} \ |f|$$

Epsilon machine



La variable `b` est divisée par deux à chaque étape tant que la somme de `a` et `b` demeure différente (`~=`) de `a`. Si on opérait sur des nombres réels, ce programme ne s'arrêterait jamais, tandis qu'ici, il s'interrompt après un nombre fini d'itérations et renvoie la valeur suivante pour `b` : `1.1102e-16`$= \epsilon_M/2$. Il existe donc au moins un nombre `b` différent de 0 tel que `a+b=a`. Ceci est lié au fait que $\mathbb{F}$ est constitué de nombres isolés ; quand on ajoute deux nombres `a` et `b` avec `b<a` et `b` plus petit que ϵ_M, on obtient toujours `a+b` égal à `a`. En MATLAB, le nombre `a+eps(a)` est le plus petit majorant strict de `a` dans $\mathbb{F}$. Donc, la somme `a+b` retourne `a` pour tout `b < eps(a)`.

$F(\beta,t, e_{min}, e_{max})$
Cardinalité finie

Et ce qu'on peut représenter exactement tous les nombres réels sur machine?

- Non , $F(\beta,t, e_{min}, e_{max}) \subset \mathbb{R}$

- Solution : Remplacer le nombre réel sur machine par un autre admettant une représentation en virgule flottante selon le système considéré.

$F(\beta,t, e_{min}, e_{max})$
Arrondis

❖Troncature ou truncating:

Ecrire le nombre et ne conserver que les *t* premiers chiffres d'une mantisse.

Exemple : $0{,}564551 \times 10^5$ →0,5645

❖Arrondi au plus proche ou *rounding to nearest ou perfect rounding* :

Substituer au nombre réel le nombre à virgule flottante qui lui est le plus proche;

Exemple : $0{,}564551 \times 10^5$ →0,5646

0,14 → 0,1 0,16 → 0,2

❖Arrondis dirigés :

Lorsque le nombre se situe à égale distance des deux nombres à virgule flottante qui l'entourent. Plusieurs alternatives sont possibles :

$F(\beta,t, e_{min}, e_{max})$
Arrondis

❖ Arrondis dirigés :

- l'arrondi par défaut ou *rounding half down* : prendre le nombre à virgule flottante le plus petit ;
- l'arrondi par excès ou rounding half up: prendre le nombre à virgule flottante le plus grand, ;
- l'arrondi vers zéro ou chopping : prendre le nombre à virgule flottante le plus petit en valeur absolue
- l'arrondi vers l'infini: le nombre à virgule flottante le plus grand en valeur absolue,
- l'arrondi au chiffre pair ou rounding half to even : prendre le nombre à virgule flottante dont le dernier chiffre de la mantisse est pair.

  0,25 →0,2 0,15 →0,2 0,05 →0
- l'arrondi au chiffre impair ou rounding half to even : prendre le nombre à virgule flottante dont le dernier chiffre de la mantisse est impair .

$F(\beta,t,e_{min},e_{max})$
Arrondis

En IEEE754 :

1.l'arrondi au plus proche, qui est celui utilisé par défaut, avec une stratégie d'arrondi au chiffre pair,

2.l'arrondi vers zéro,

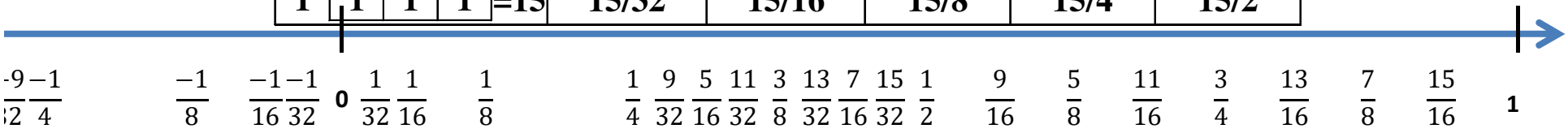
3 et 4. les arrondis vers $\pm\infty$)

$F(\beta,t,e_{min},e_{max})$
Arrondis

Perfect rounding : 3,4 →3,5 et 0,94 → 1

Rounding half toward zero : 3,4 →3 0,94 → 0,875= $\frac{7}{8}$

$f = n \times 2^{e-t}$									
n					2^{e-t}				
					$e=-2$	$e=-1$	$e=0$	$e=1$	$e=2$
					1/32	1/16	1/8	1/4	1/2
1	0	0	0	=8	1/4	1/2	1	2	4
1	0	0	1	=9	9/32	9/16	9/8	9/4	9/2
1	0	1	0	=10	5/16	5/8	5/4	5/2	5
1	0	1	1	=11	11/32	11/16	11/8	11/4	11/2
1	1	0	0	=12	12/32	3/4	3/2	3	6
1	1	0	1	=13	13/32	13/16	13/8	13/4	13/2
1	1	1	0	=14	14/32	7/8	7/4	7/2	7
1	1	1	1	=15	15/32	15/16	15/8	15/4	15/2



$F(\beta, t, e_{min}, e_{max})$
Arrondis

```
> x=0.1
x =
    0.1000

>> x+0.2==0.3
ans =
  logical
   0
```

```
>>> a = 0.1
>>> a
1.0000000000000006e-01
>>> b = 0.2
>>> b
2.0000000000000011e-01
>>> c = 0.1 + 0.2
>>> c
3.0000000000000044e-01
>>> d = 0.3
>>> d
2.9999999999999989e-01
```

L'opérateur float : fl(x)

- Pour décrire l'ensemble des nombres réels qui trouvent une représentation en virgule flottante sur F, on définit l'ensemble

$$G = \quad \{x \in \mathbb{R} \text{ tel que } \quad val\_min \leq |x| \leq val\_max\} \cup \{0\}$$
$$G = \quad [-(2^{-t} - 2) \times 2^{e_{max}}, -2^{e_{min}}] \cup \{0\} \cup [2^{e_{min}}, (2 - 2^{-t}) \times 2^{e_{max}}]$$

- Et l'opérateur $fl\ (x)$: l'arrondi au plus proche d'un nombre réel x, définissant ainsi l'application :

$$fl: \quad G \rightarrow \quad \mathrm{F}(\beta, t,\ e_{min}, e_{max})$$
$$x \rightarrow \quad f, \text{ plus proche } f \ \in \mathrm{F}$$

- La valeur de l'arrondit est décidé comme suit :

$$fl(x) = (-1)^s(1.a_1a_2 \ldots\ldots\ \hat{a}_t\ )\beta^e, \qquad \hat{a_t} = \begin{cases} a_t & \text{si } a_{t+1} < \beta/2 \\ a_t + 1 & \text{si } a_{t+1} \geq \beta/2 \end{cases}$$

Récapitulatif

- L'ensemble des réels présente quelques propriétés :
  - il est non borné,
  - on trouve
    toujours un réel entre deux autres réels,
  - il contient un nombre infini d'éléments.

- Alors que pour les flottants F
  - il existe un nombre fini de mantisses différentes, car elles sont codées sur un nombre fini de *bits*;
    il existe pour la même raison un nombre fini d'exposants, et donc un nombre fini de flottants;
  - un nombre fini d'exposant signifie un exposant maximal et minimal, c'est-à-dire des bornes à l'ensemble des flottants en les associant respectivement aux mantisses maximales et minimales.

Mesure de l'erreur

- **<u>l'erreur absolue</u>** est définie entre un scalaire réel x et son approximation $\hat{x}$ par : $|x - \hat{x}|$,

- **<u>l'erreur relative</u>** utilisée lorsque x est non nul, est donnée par $\frac{|x-\hat{x}|}{|x|}$.

Mesure de l'erreur

- Exemple

Soit la valeur $X = \frac{1}{7}$.

Sa valeur approchée représentée sur 6 chiffres est X=0,142857

Sa valeur exacte est $\bar{X} = \frac{1}{7} = 0,\overline{142857}$

$$\Delta X = |X - \bar{X}| = |0.142857 - \frac{1}{7}| = \frac{1}{7} * 10^{-6}$$

$$\rho(X) = |\frac{X - \bar{X}}{\bar{X}}| = 10^{-6} = 0.0001\%$$

Erreur sur fl(x)

Prenons un flottant f de la forme:

$$f = s \times 2^e$$

Le successeur direct de f, f^+, est obtenu en incrémentant le dernier *bit* du significande de f. Son significande s^+ est ainsi égal à:

$$s^+ = s + 2^{-52}$$

Le prédécesseur direct de f, f^-, est obtenu de manière similaire, mais en décrémentant le dernier *bit* du significande de f. Son significande s^- vaut alors:

$$s^- = s - 2^{-52}$$

On peut alors utiliser la formule du paragraphe précédent pour calculer l'erreur d'arrondi maximale:

$$|max| = \frac{s - s^-}{2} \times 2^e = \frac{s^+ - s}{2} \times 2^e = 2^{-53} \times 2^e$$

On a donc une erreur d'arrondi d'au pire 2^{-53} sur le significande. L'erreur sur le flottant dans son ensemble dépend de l'exposant e et vaut au plus 2^{e-53}.

Propagation des erreurs

- Les formules suivantes nous donnent les erreurs qu'on obtient lorsque l'on effectue des opérations arithmétiques sur des valeurs connues avec une précision limité
- Pour l'erreur absolue:
  - $\Delta(x \pm y) = \Delta x + \Delta y$
  - $\Delta xy = |x|\Delta x + |y|\Delta y$
  - $\Delta({}^{x}/_{y}) = \frac{y\Delta x + x\Delta y}{y^2}$

- Pour l’erreur relative:
  - $E_{(x+y)} = \frac{\Delta x+\Delta y}{|x+y|}$
  - $E_{(x-y)} = \frac{\Delta x+\Delta y}{|x-y|}$
  - $E_{x.y} = E_x + E_y$
  - $E_{x/y} = E_x + E_y$