0/

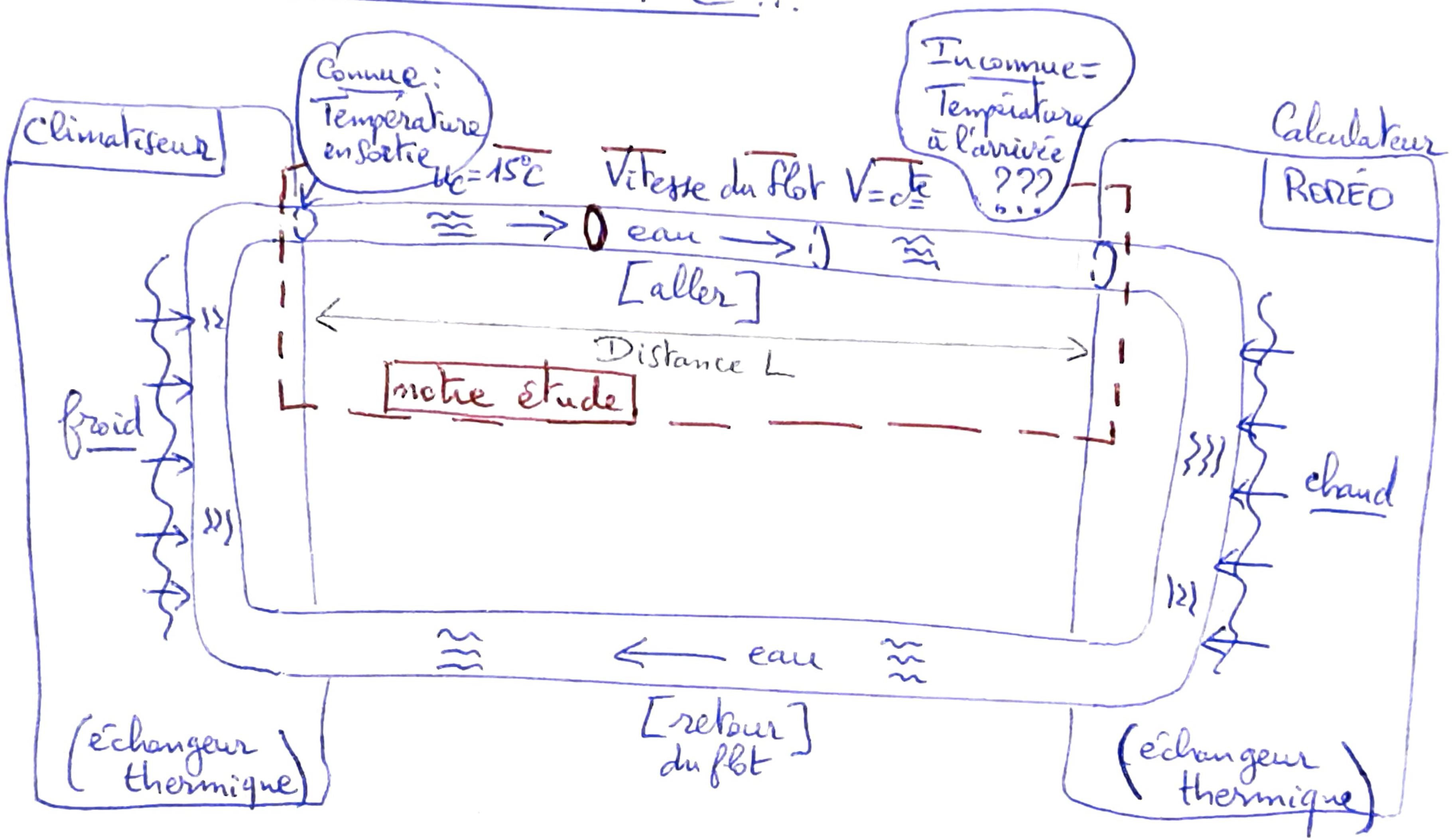
PROJET

CLIMATISATION de ROMEO

SUIVI D'UN FLOT D'EAU DANS LA CONDUITE

ETUDE DE LA DEPERDITION DE CHALEUR

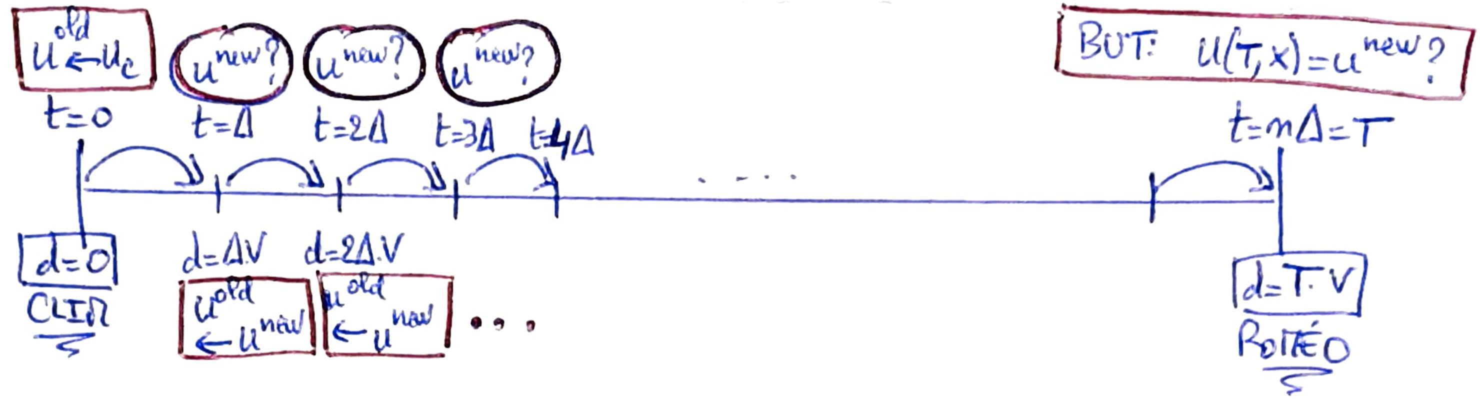
Contexte = !!! Canicule => $u_{Ext} = 40°C$!!!



BUT = Suivre l'évolution de la température du flot d'eau se propageant dans la section 2D : Ω de la conduite du climatiseur, de la sortie du climatiseur ... jusqu'à l'arrivée au calculateur.

... la sortie du climatiseur ... jusqu'à l'arrivée au calculateur.

Propagation Itérative (= schéma qui propage la condition initiale) en n pas



OBJECTIF = amener l'eau du flot à l'arrivée de Roméo à température moyenne $\bar{u}(T) \simeq 20°C$, tout en faisant le plus "d'économies pour la planète"...

paramètres fixes:

$C = 4{,}2 \cdot 10^6$ (constante calorifique)
$K_{eau} = 0{,}6 \ Wm^{-1}K^{-1}$ (conductivité thermique de l'eau)
$f = 0$ (source de chaleur)
$R = 0{,}05$ m (rayon de la section)
$L = 50$ m (longueur de la conduite)
$u_E = 40°C$ (température extérieure)
$n = 10$ (nombre d'itération du schéma)

Vos "manettes" :-)

paramètres de réglage:
(Valeurs données à titre d'exemple)

↗ $V = 0{,}5$ Km/h $= (0{,}5/3{,}6)$ m/s (vitesse du flot)

↘ (en isolant) $\alpha = \alpha_{Inox} = 26/0{,}002 = \dfrac{K_{Inox}}{e_{Inox}}$ (transfert)

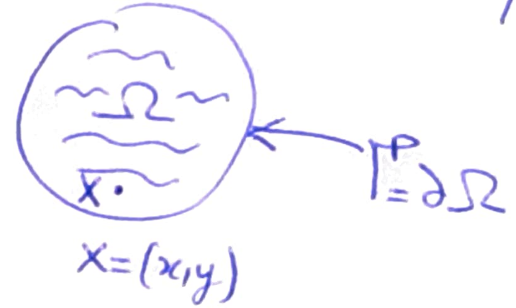
↗↘ $u_c = 15°C$ (température Clim)

paramètres inter-dépendants:

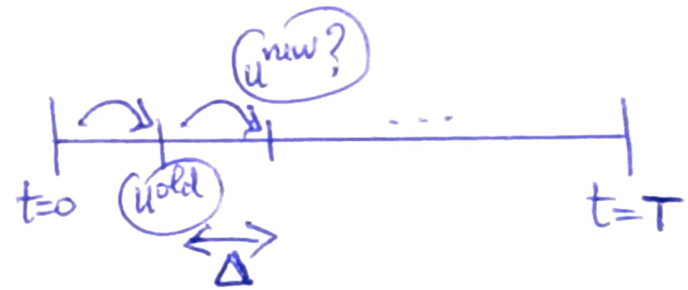
$T = L/V$ (temps de parcours du flot)
$\Delta = T/n$ (pas de temps dans le schéma)

B/ Trouver $u(t,x)$? /

$$\begin{cases} u(t=0,x) = u_C \quad (\text{temp. de Clim.}), \ \forall x \in \overline{\Omega} \\ c\dfrac{\partial u}{\partial t} - \nabla\cdot(K\nabla u) = f(x), \ \forall x=(x,y)\in\Omega, \ \forall t\in]0,T] \\ -K\dfrac{\partial u}{\partial \vec{n}}(t,x) = \alpha\left[u(t,x)-u_E\right], \ \forall x\in\Gamma, \ \forall t\in]0,T]. \end{cases}$$



Discrétisation en Temps (Différences Finies)



$\boxed{\text{EULER IMPLICITE}}$

$$c\left(\frac{u^{new}-u^{old}}{\Delta}\right) - \nabla\cdot(K\nabla u^{new}) = f$$

Formulation Variationnelle (pour EF !)

$$\iint_\Omega \left[c\left(\frac{u^{new}-u^{old}}{\Delta}\right) - \nabla\cdot(K\nabla u^{new}) = f \right] \times v \, dx\,dy \quad \left(\begin{array}{c}\text{pour toute fonction}\\ \text{test } v\in V_h\end{array}\right)$$

Int$^{\circ}$ = par parties

$$\iint_\Omega c\left(\frac{u^{new}-u^{old}}{\Delta}\right) v\, dx\,dy + \iint_\Omega K\nabla u^{new}\cdot\nabla v \, dx\,dy + \int_\Gamma \alpha(u^{new}-u_E)\, v\, ds = \iint_\Omega f v \, dx\,dy$$

<u>réorganisation des termes</u> [u^{new} : inconnue , u^{old} : connu] bilinéaire / linéaire

$$\underbrace{\frac{c}{\Delta}\iint_\Omega u^{new}\times v\,dxdy + \underbrace{\iint_\Omega k\nabla u^{new}\cdot\nabla v\,dxdy + \int_\Gamma \alpha u^{new}\times v\,ds}_{a(u^{new},v)}}_{a^{inst}(u^{new},v)} = \underbrace{\frac{c}{\Delta}\iint_\Omega u^{old}\times v\,dxdy + \underbrace{\iint_\Omega f v\,dxdy + \int_\Gamma \alpha u_E v\,ds}_{\ell(v)}}_{\ell^{inst}(v)}$$

$\boxed{\text{Rem}}$:

$$\begin{cases} a^{inst.}(u,v) = a(u,v) + \dfrac{c}{\Delta}\displaystyle\iint_\Omega u\,v\,dxdy \\ \ell^{inst.}(v) = \ell(v) + \dfrac{c}{\Delta}\displaystyle\iint_\Omega u^{old}\,v\,dxdy \end{cases}$$

<u>formes</u> : <u>bilinéaire</u>

: <u>linéaire</u>

<u>cas instationnaire</u>

<u>cas stationnaire</u> [cf CM1]

termes instationnaires (supplémentaires)

⚠ $\ell^{inst}(\cdot)$ dépend de u^{old}, il faudra remettre à jour cette forme linéaire à chaque pas de temps.

En remplaçant la fonction v par n'importe quelle fonction de base φ_I chapeau ($1 \leq I \leq N = mbn$) 5/
et en décomposant $\boxed{u^{new/old} = \sum_{J=1}^{N} u_J^{new/old} \varphi_J}$ dans la base des fonctions chapeaux,
pour former les vecteurs $U^{new/old} = \begin{pmatrix} u_1^{new/old} \\ \vdots \\ u_N^{new/old} \end{pmatrix} \in \mathbb{R}^N$, on obtient
l'écriture matricielle :

$$\frac{c}{\Delta} \mathbb{M} U^{new} + \mathbb{A} U^{new} = \mathbb{F} + \frac{c}{\Delta} \mathbb{M} U^{old}$$

avec $\left[\mathbb{A} = \left(a(\varphi_J, \varphi_I)\right)_{1 \leq I,J \leq N},\ \mathbb{M} = \left(\iint_\Omega \varphi_J \varphi_I \, dx\right)_{1 \leq I,J \leq N},\ \mathbb{F} = \left(\ell(\varphi_I)\right)_{1 \leq I \leq N} \right]$

soit encore :

$$\boxed{\begin{aligned} &U^{new} ? \,/\, \left(\frac{c}{\Delta}\mathbb{M} + \mathbb{A}\right) U^{new} = \mathbb{F} + \frac{c}{\Delta} \mathbb{M} U^{old} \\ &U^{new} ? \,/\, \underbrace{\mathbb{B}}_{\text{matrice } N \times N} \cdot U^{new} = \underbrace{\mathbb{G}^{old}}_{\text{vecteur de } \mathbb{R}^N} \end{aligned}}$$

Algo ... Rem : on va faire un certain nombre d'itér... 6/

matrice $N \times N$ vecteur de $\mathbb{R}^N$

$\boxed{\text{Algorithme Itératif}}$

<u>Rem</u> : on va faire un certain nombre d'itérations ($n = 10$ par ex.) [autant que nécessaire pour que le flot d'eau atteigne ROMEO]

```
U^old ← u_c × (1 ... 1) ∈ R^N    % : instant initial
t ← 0.0
pour  i ← 1:n        % : i = indice de l'instant t = iΔ.
      t ← t+Δ        % : ou bien t ← i×Δ !            (nouvel instant)
      d ← t×V        % : distance à la climatisation  (nouvelle position)
      G^old ← F + (c/Δ) M U^old   % : Matrice × Vecteur + Vecteur ≡ "Gaxpy".
      U^new ? / B U^new = G^old   % : Résolution d'un système linéaire
      U^old ← U^new
fin (pour i)
```

$$U^{old} \leftarrow u_c \times \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^N \quad \% : \text{instant initial}$$

<u>Rem</u> : <u>Les Matrices (+Vecteurs)</u> : $\mathbb{A}$, $\mathbb{M}$, $\mathbb{F}$, $\mathbb{B} = \frac{c}{\Delta}\mathbb{M} + \mathbb{A}$ peuvent être <u>assemblés</u> $\boxed{\text{avant}}$ cet algorithme itératif, <u>une fois pour toute</u> (ils sont <u>indépendants</u> de l'itération)