Modélisation de l'évolution de la température

Approche par un automate cellulaire.

Huet Natanéo Hurot Eliott

2024

Problématique

Comment modéliser l'évolution de la température d'une maison via un automate cellulaire ?

Sommaire

Définition d'automate cellulaire

4-uplet (d, Q, V, δ)

d est la dimension

Q est l'alphabet

V est le voisinage

$\delta : Q^{\alpha} \longrightarrow Q$ est la règle de transistion

Premier modèle

$$\Delta U = W + Q$$
$$C_v \Delta T_{cellule} = W + Q$$
$$\delta Q = \phi \Delta t$$
$$\Delta T_{difference} = \phi R_{th}$$
$$\Delta T_{cellule} = \frac{\Delta T_{difference} \times \Delta t}{R_{th} \times C_v}$$

Définition des structures

```
1  struct cell
2  {
3      float T; // --------------- en degre Celsius
4      char* type; // ------------ type de la cellule (airExt, airInt, wall, window, door)
5      coord co; // -------------- coordonnees de la cellule
6      float CTherVol; // -------- Capacite thermique volumique en J.K-1.m-3
7      float lambda; // ---------- Conductivite thermique en W.K-1.m-1
8      float surface; // --------- Surface en m2
9      float epaisseur; // ------- Epaisseur en m
10     float lambda_iso_ext; // -- Conductivite thermique de l'isolant exterieur en W.K-1.m
        -1
11     float lambda_iso_int; // -- Conductivite thermique de l'isolant interieur en W.K-1.m
        -1
12     float epaisseur_iso_ext; // Epaisseur de l'isolant exterieur en m
13     float epaisseur_iso_int; // Epaisseur de l'isolant interieur en m
14 };
15
```

Définition des structures

```
1  struct cell
2  {
3      float T; // --------------- en degre Celsius
4      char* type; // ------------ type de la cellule (airExt, airInt, wall, window, door)
5      coord co; // -------------- coordonnees de la cellule
6      float CTherVol; // -------- Capacite thermique volumique en J.K-1.m-3
7      float lambda; // ---------- Conductivite thermique en W.K-1.m-1
8      float surface; // --------- Surface en m2
9      float epaisseur; // ------- Epaisseur en m
10     float lambda_iso_ext; // -- Conductivite thermique de l'isolant exterieur en W.K-1.m
        -1
11     float lambda_iso_int; // -- Conductivite thermique de l'isolant interieur en W.K-1.m
        -1
12     float epaisseur_iso_ext; // Epaisseur de l'isolant exterieur en m
13     float epaisseur_iso_int; // Epaisseur de l'isolant interieur en m
14 };
15
```

Exemple de structure

```
structure window1 = {
    .T = 10.0,
    .type = "window",
    .begining = {.i = 2, .j = 8},
    .ending = {.i = 2, .j = 10},
    .CTherVol = 17000,
    .lambda = 1.0,
    .surface = 2.0,
    .epaisseur = 0.1,
};

```

Règles de l'automate cellulaire

L'air ext est constant.

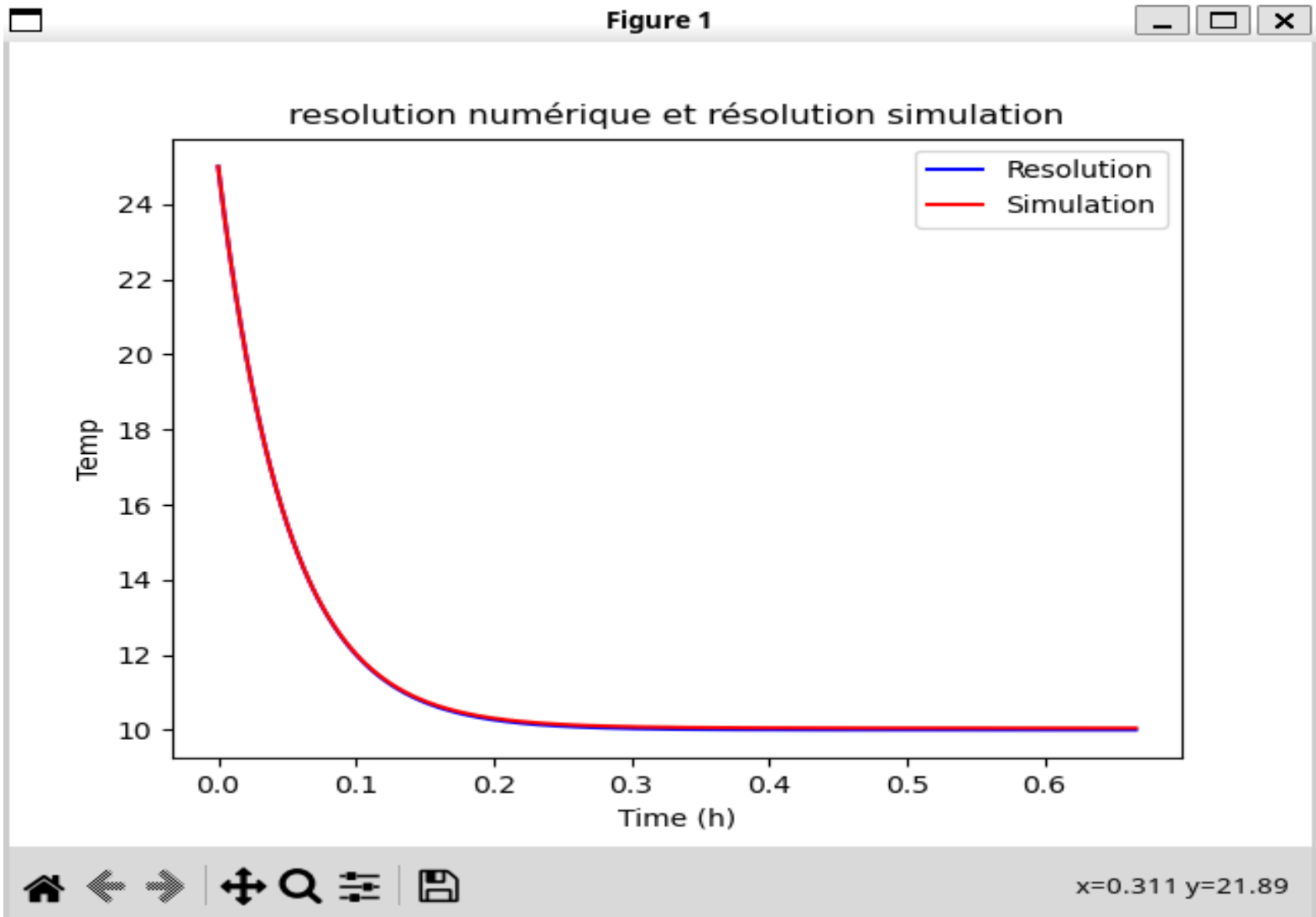
Pour tout le reste des cellules du modèle on calcule Q en fonction des 4 cellules adjacentes à celle que l'on regarde.

Pour l'air int on calcule Q avec les résistances termiques des cellules d'air adjacentes.

Fonction de Transfert

```
100  cell* nextStep(cell* map, int height, int width){
101      cell* newMap = malloc(sizeof(cell) * height * width);
102      int toAdd[4][2] = {{0, 1}, {0, -1}, {1, 0}, {-1, 0}}; // {i, j}
103      for (int i = 0; i < height; i++){
104          for (int j = 0; j < width; j++){
105              cell actCell = map[i*width + j];
106              newMap[i*width + j] = map[i*width + j];
107              if (strcmp(actCell.type, "airInt") == 0){
108                  float Q = 0;
109                  for (int l = 0; l<4; l++){
110                      cell c = map[(i + toAdd[l][0])*width + j + toAdd[l][1]];
111                      float rth = c.epaisseur / (c.lambda * c.surface);
112                      Q += (c.T - actCell.T) / (rth * (actCell.surface * actCell.CTherVol));
113                  }
114                  newMap[i*width + j].T += Q;
115              } else if (strcmp(actCell.type, "wall") == 0){
116                  for (int l = 0; l<4; l++){
117                      cell c = map[(i + toAdd[l][0])*width + j + toAdd[l][1]];
118                      if (strcmp(c.type, "airExt") == 0){
119                          newMap[i*width + j].T = c.T;
120                          break;
121                      }
122                  }
123              }
124          }
125      }
126      free(map);
127      return newMap;
128  }
```

Validation du premier modèle



Problème soulevé

Heure : 24

10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	12.2	13.5	13.5	12.2	10.0	10.0	10.0	10.0
10.0	10.0	10.0	13.5	15.5	15.5	13.5	10.0	10.0	10.0	10.0
10.0	10.0	10.0	13.5	15.5	15.5	13.5	10.0	10.0	10.0	10.0
10.0	10.0	10.0	12.2	13.5	13.5	12.2	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Solution

Présence d'un ventilateur homogénisant la pièce : 1

```
if (strcmp(actCell.type, "airInt") == 0){
    float Q = 0;
    float convection = 0;
    for (int l = 0; l<4; l++){
        cell c = map[(i + toAdd[l][0])*width + j + toAdd[l][1]];
        float rth = c.epaisseur / (c.lambda * c.surface);
        Q += (c.T - actCell.T) / (rth * (actCell.surface * actCell.CTherVol));
        convection += c.T - actCell.T;
    }
    newMap[i*width + j].T += Q + convection/(4*300);
```

Figure: Fonction nextStep mise à jour

Résultats

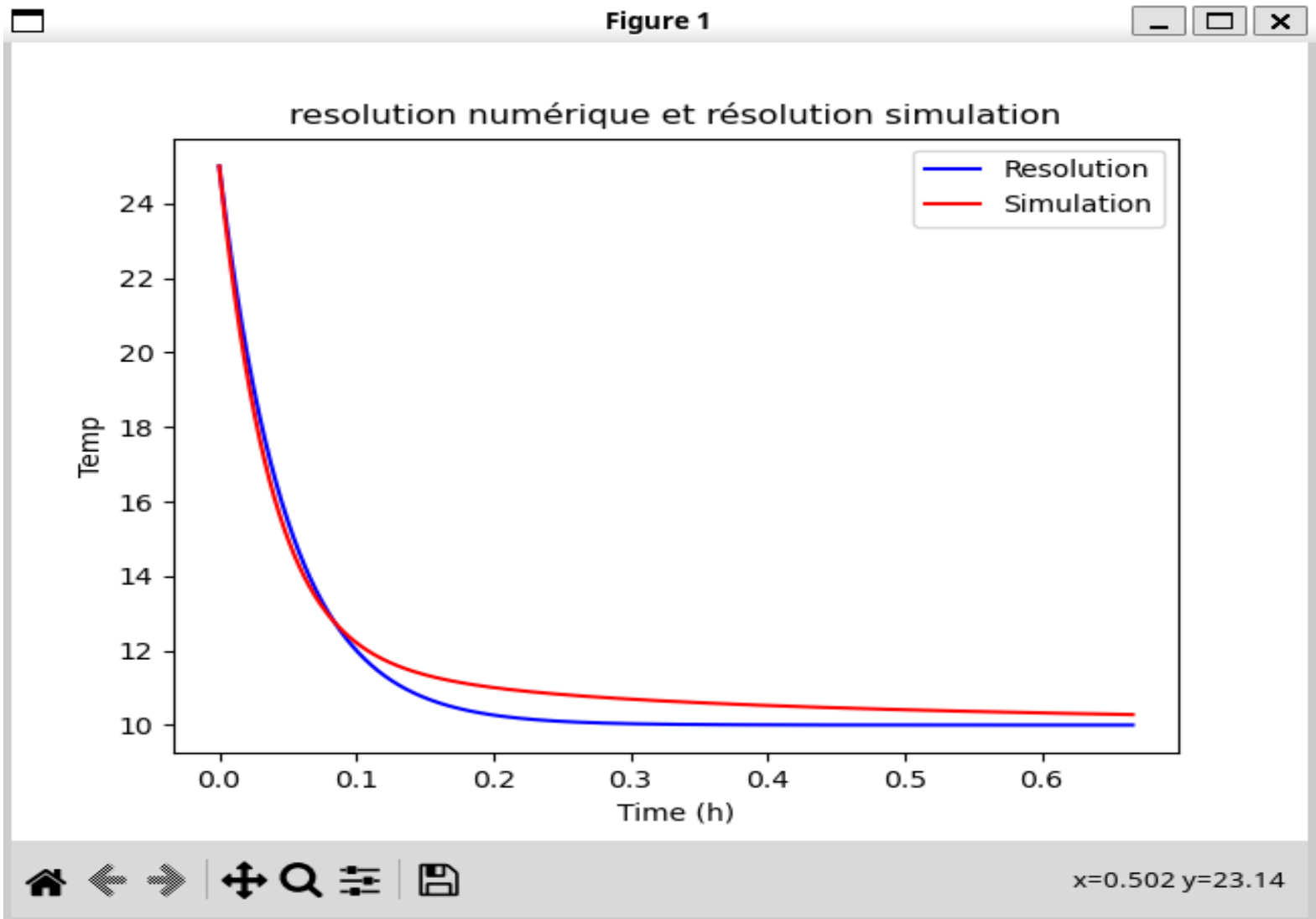


Figure 1
resolution numérique et résolution simulation
Resolution
Simulation
Temp
Time (h)
24
22
20
18
16
14
12
10
0.0
0.1
0.2
0.3
0.4
0.5
0.6
x=0.502 y=23.14

```
Heure : 24
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
```

Modèle final

$$\Delta T_{cellule} = \frac{\Delta T_{difference} \times \Delta t}{R_{th} \times C_v}$$

Ici, on différencie l'isolation intérieur de l'extérieur.
Cela permet aussi de donner aux murs leurs propre température.

On note aussi l'ajout d'autres structures telles que les portes ou les fenêtres, ayant leurs propres C_v, λ, etc.

1h :

10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.1	10.2	10.3	10.3	10.1	10.0	10.0	10.0	10.1	10.0	10.0	10.0	10.0
10.0	10.0	10.0	11.2	11.0	10.3	10.2	10.3	10.5	10.7	10.8	11.2	10.4	10.2	10.2	10.2	10.4	10.0	10.0	10.0
10.0	10.0	10.0	11.5	11.4	10.4	10.2	10.5	12.0	12.9	13.0	12.4	10.7	10.3	10.3	10.5	10.5	10.0	10.0	10.0
10.0	10.0	10.0	11.8	11.7	10.5	10.3	10.7	13.2	14.6	14.7	13.5	11.6	10.7	10.5	10.6	10.6	10.0	10.0	10.0
10.0	10.0	10.0	12.0	12.0	10.7	10.4	11.1	14.1	15.7	15.7	13.9	10.9	10.3	10.3	10.4	10.4	10.0	10.0	10.0
10.0	10.0	10.0	12.0	12.1	11.4	10.8	12.5	14.7	16.1	15.9	14.1	10.9	10.2	10.2	10.3	10.3	10.0	10.0	10.0
10.0	10.0	10.1	11.5	11.8	11.3	10.7	12.3	14.4	15.7	15.5	13.9	11.7	10.5	10.3	10.3	10.3	10.0	10.0	10.0
10.0	10.0	10.2	10.5	11.0	10.5	10.4	10.9	13.1	14.5	14.4	12.9	10.7	10.3	10.3	10.4	10.3	10.1	10.0	10.0
10.0	10.0	10.1	10.2	10.6	10.3	10.2	10.4	11.5	12.9	12.8	11.4	10.4	10.2	10.2	10.4	10.2	10.1	10.0	10.0
10.0	10.0	10.1	10.2	10.4	10.2	10.1	10.2	10.6	11.6	11.6	10.6	10.2	10.1	10.2	10.4	10.2	10.1	10.0	10.0
10.0	10.0	10.1	10.1	10.4	10.2	10.2	10.3	11.2	12.5	12.5	11.2	10.3	10.2	10.2	10.6	10.3	10.1	10.0	10.0
10.0	10.0	10.1	10.2	10.4	10.2	10.2	10.5	12.2	13.4	13.4	12.2	10.5	10.2	10.3	10.9	10.8	10.1	10.0	10.0
10.0	10.0	10.1	10.2	10.4	10.3	10.2	10.5	12.6	13.9	13.9	12.6	10.6	10.2	10.3	11.0	11.1	10.0	10.0	10.0
10.0	10.0	10.1	10.5	10.5	10.3	10.2	10.5	12.5	13.7	13.7	12.5	10.5	10.2	10.3	11.0	11.1	10.0	10.0	10.0
10.0	10.0	10.0	10.5	10.5	10.3	10.2	10.5	11.9	12.8	12.8	11.8	10.5	10.2	10.3	10.8	10.9	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.5	10.0	10.0	10.0	10.4	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Programmation

```
else if (strcmp(actCell.type, "isolated wall") == 0)
    {
        float Q = 0;
        for (int l = 0; l < 4; l++){

            if (i+toAdd[l][0] >= 0 && i+toAdd[l][0] < height && j+toAdd[l][1] >= 0 && j+toAdd[l][1] < width){
                cell c = map[(i+toAdd[l][0])*width + j+toAdd[l][1]];

                if (strcmp(c.type, "airInt") == 0)
                {
                    Q += (c.T - actCell.T) * 1 / ((actCell.epaisseur_iso_int / (actCell.lambda_iso_int * actCell.surface)) * actCell.CTherVol * actCell.surface * actCell.epaisseur);
                }
                else if (strcmp(c.type, "airExt") == 0)
                {
                    Q += (c.T - actCell.T) * 1 / ((actCell.epaisseur_iso_ext / (actCell.lambda_iso_ext * actCell.surface)) * actCell.CTherVol * actCell.surface * actCell.epaisseur);
                }
                else
                {
                    Q += (c.T - actCell.T) * 1 / ((c.epaisseur / (c.lambda * c.surface)) * actCell.CTherVol * actCell.surface * actCell.epaisseur);
                }
            }

        }

```

Observation

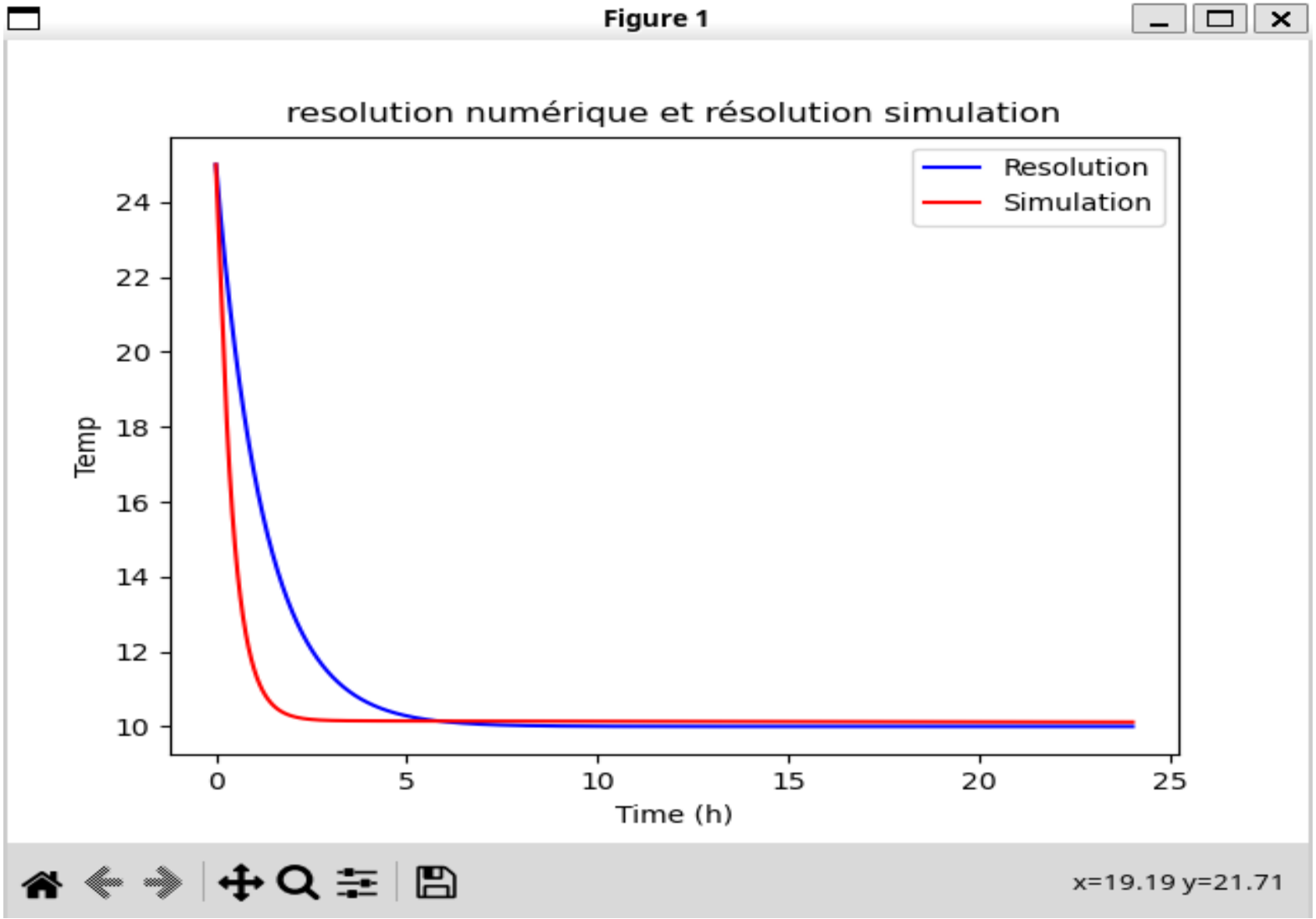


Figure: Température initial des murs à 10°C

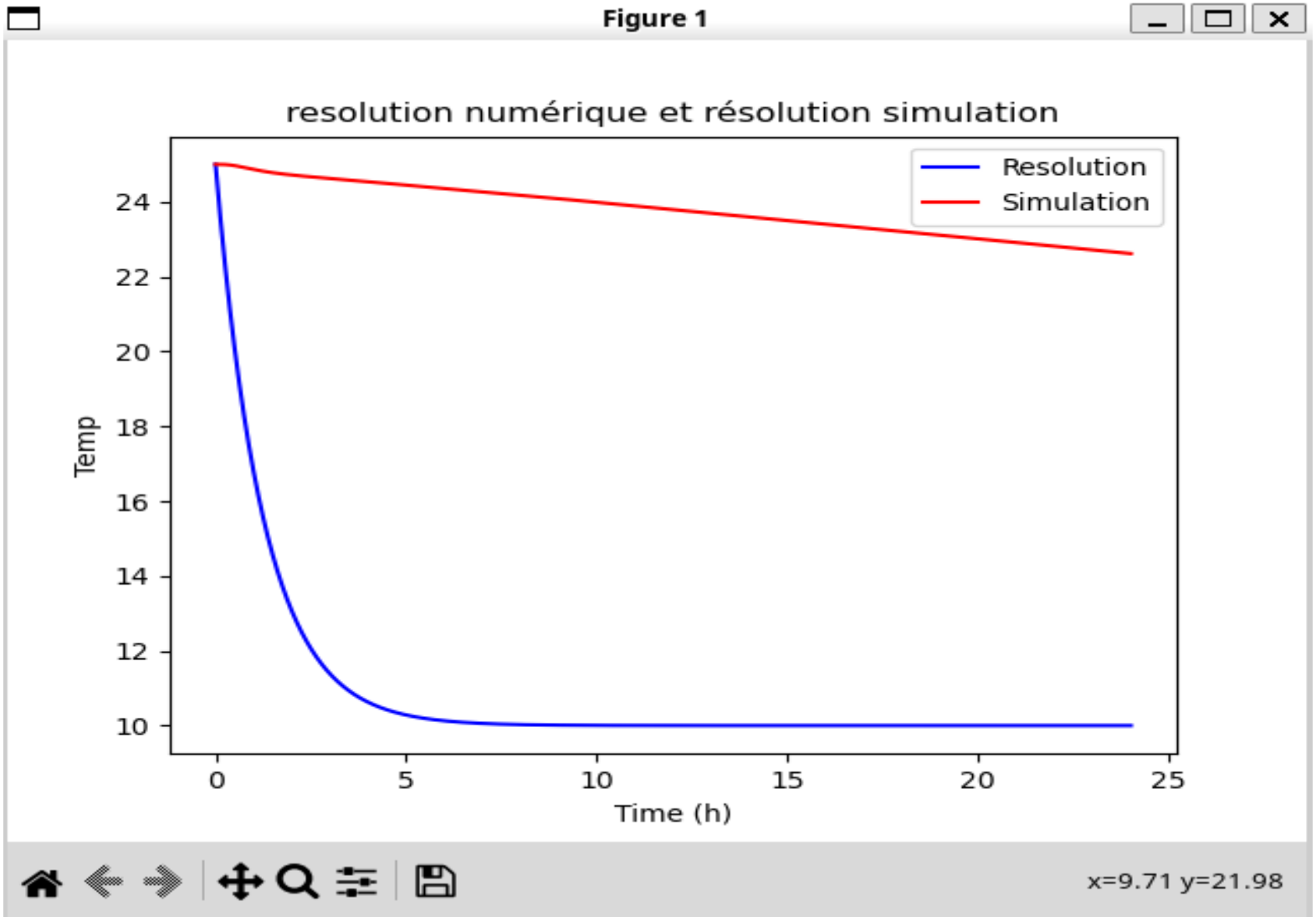


Figure: Température initial des murs à 25°C

Ajout du chauffage et autres structures

$$\begin{cases} \Delta T_{cellule} = \frac{P \times \Delta t}{C_v} \text{ si } T_{cellule} < 18\text{C} \\ \text{Aucun chauffage sinon} \end{cases}$$

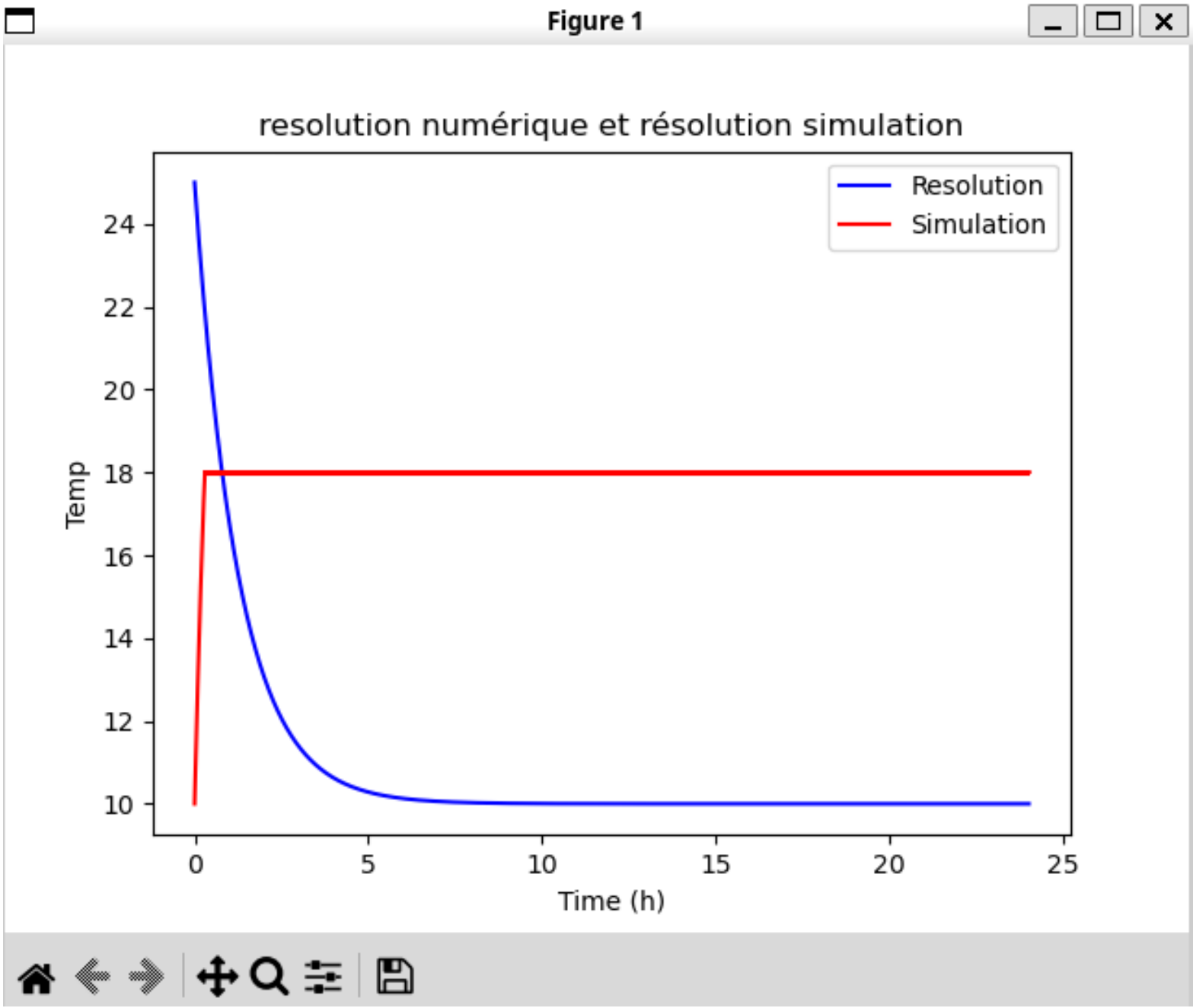


Figure 1
resolution numérique et résolution simulation
Resolution
Simulation
Temp
Time (h)
24
22
20
18
16
14
12
10
0
5
10
15
20
25

Objectifs

Réduire les suppositions (loi de newton, pont thermique...)

Ajout des plages de chauffage

Ajout d'une troisième dimension

Précision de la problématique