Modélisation des îlots de chaleur urbains

Présentation TIPE réalisée par : ORKHIS Walid

Numéro d'inscription SCEI : 51523

Sommaire

Introduction

Îlot de chaleur urbain

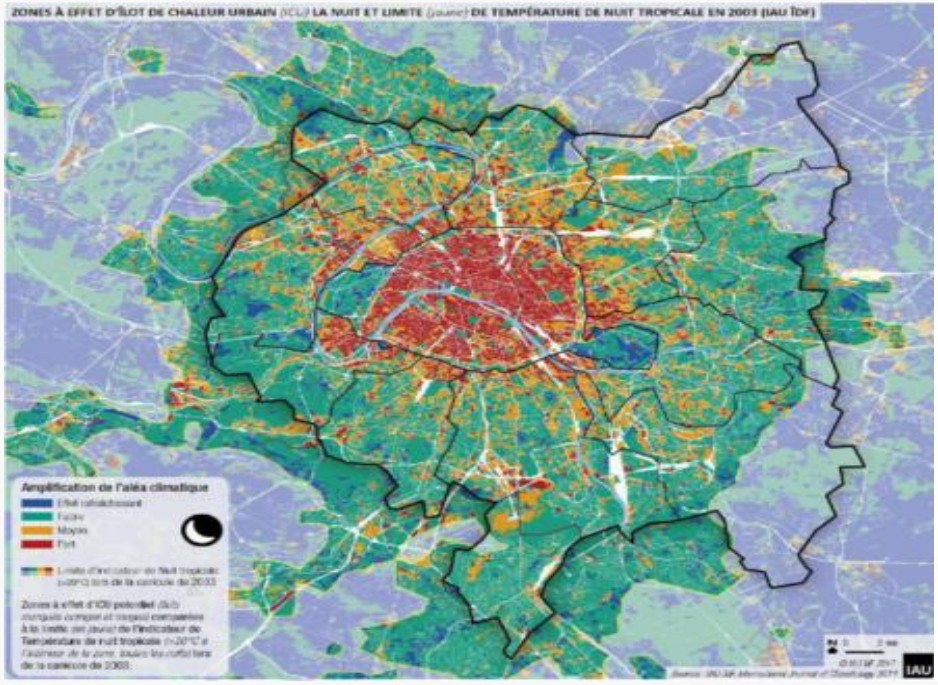


Figure : interpolation spatiale et cartographie Paris IDF

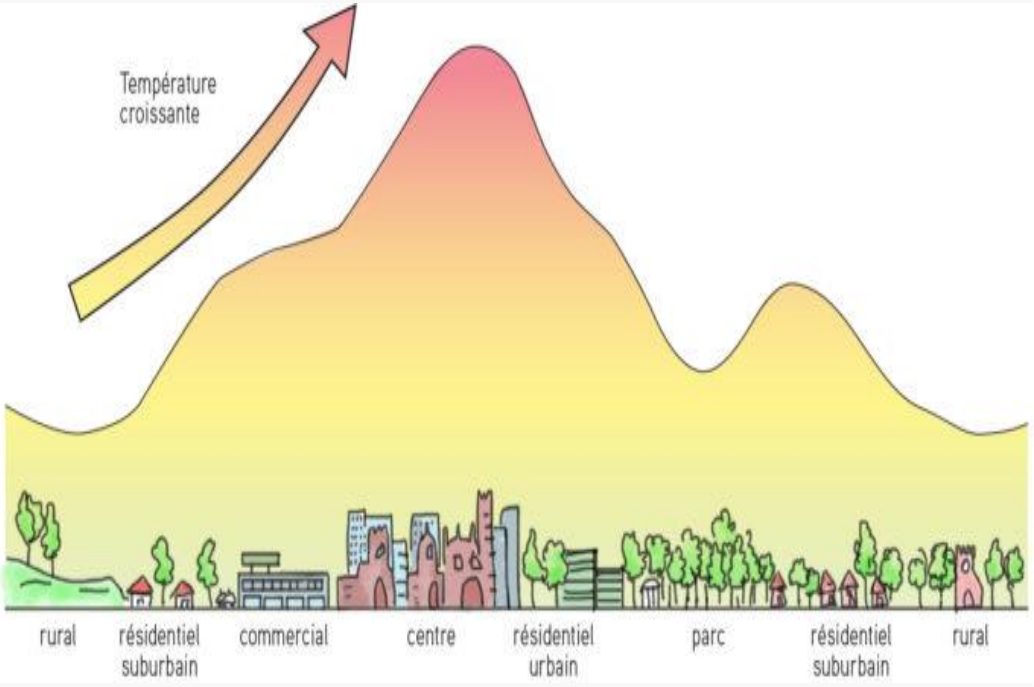


Figure :Profil d'un îlot de chaleur urbain

Modèle urbain

Canyon urbain

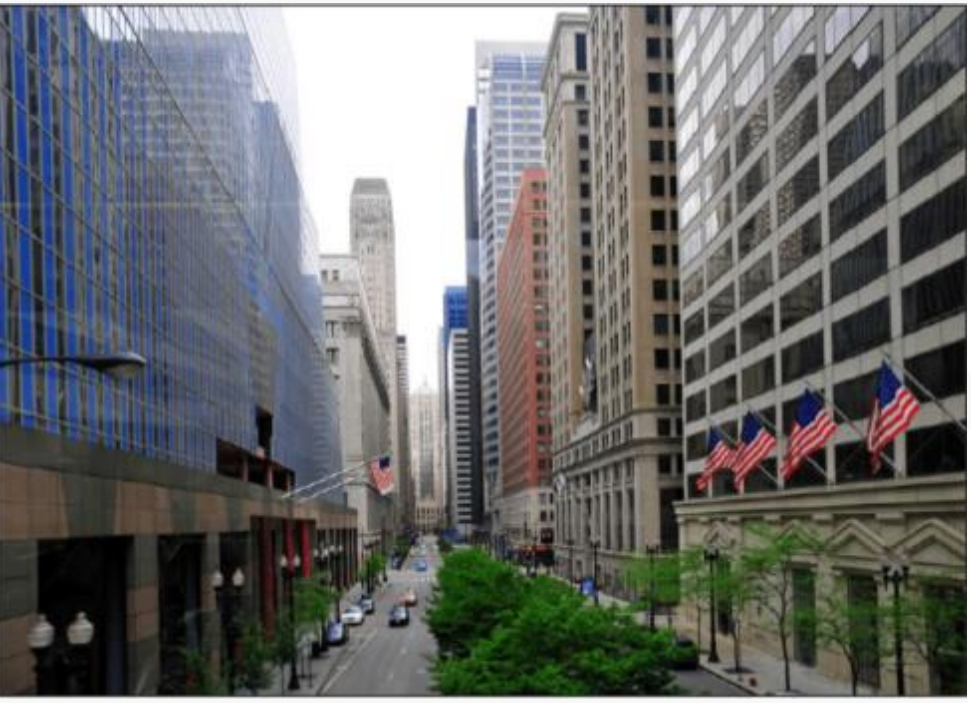


Figure 3 : **canyon urbain**

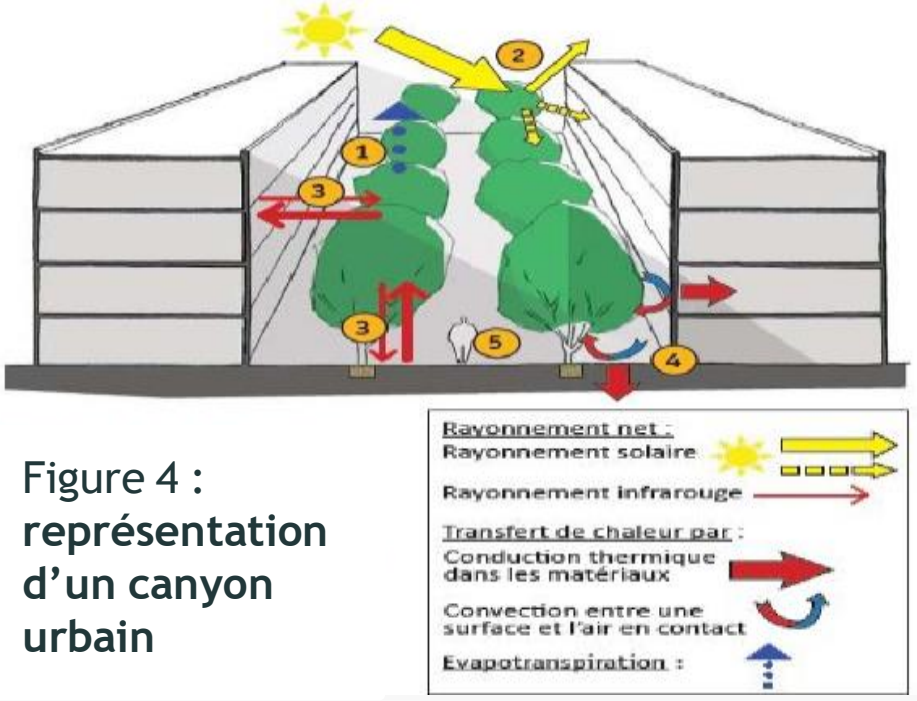
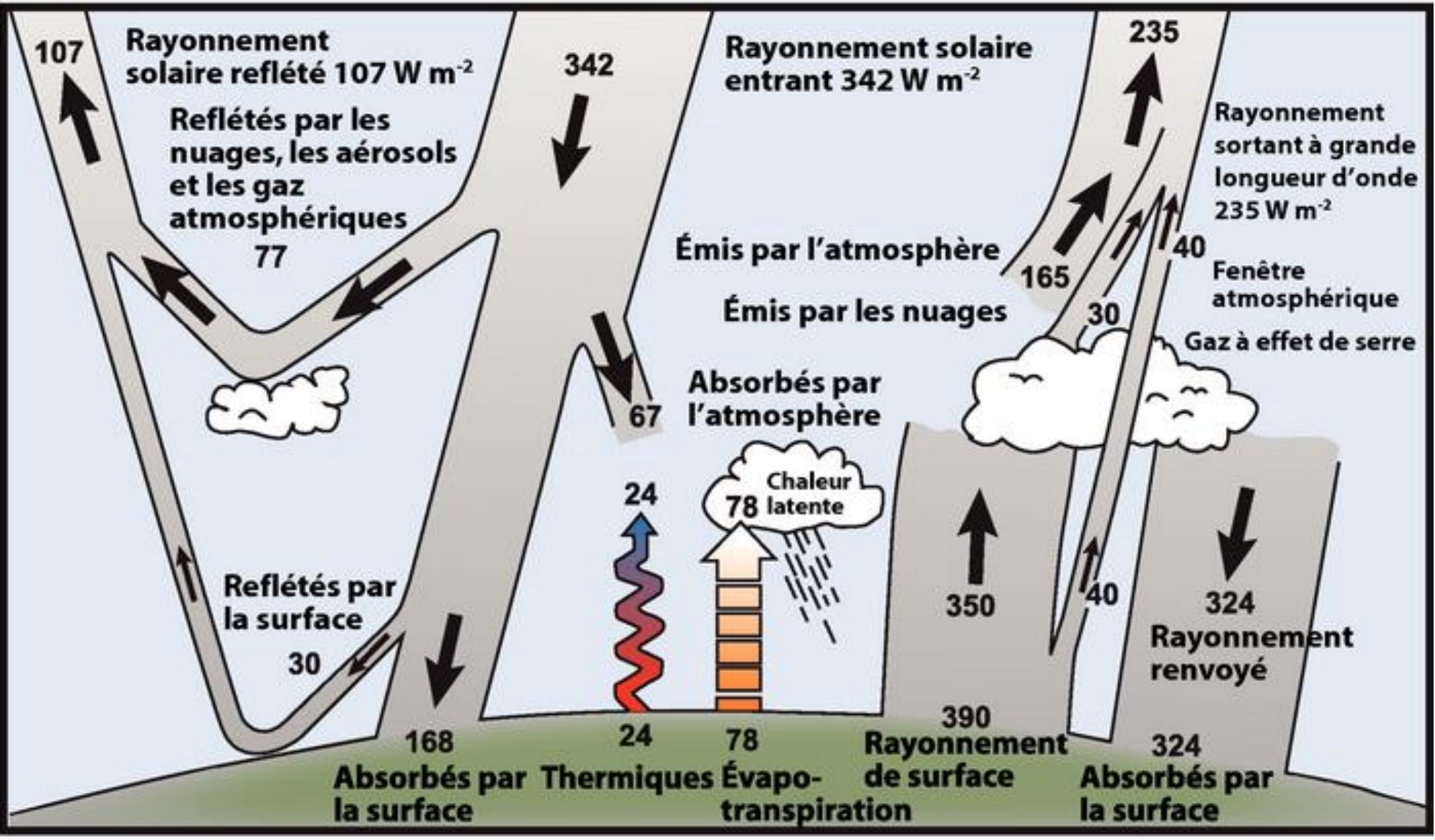


Figure 4 : **représentation d'un canyon urbain**

Bilan énergétique.

Bilan énergétique.



Bilan energetique

Le bilan énergétique de la canyon urbain peut être exprimé par l'équation suivante:

$$R_n = H + LE + G + S$$

Avec:

Rn :est le rayonnement net entrant dans la canopée (W/m^2).
H :est le flux de chaleur sensible Flux de chaleur du sol : C'est le flux de chaleur qui résulte de l'échange de chaleur entre les surfaces des bâtiments. (W/m^2)
LE :Flux de chaleur latente C'est le flux de chaleur qui résulte de l'évaporation de l'eau à la surface des bâtiments et de la transpiration des plantes dans le canyon urbain. (W/m^2)
G :Flux de chaleur du sol : C'est le flux de chaleur qui résulte de l'échange de chaleur entre la surface du sol et l'atmosphère. (W/m^2)
S :Stockage de chaleur : C'est la quantité de chaleur stockée dans les matériaux de surface des bâtiments, tels que les murs, les toits et les routes. (W/m^2)

Rayonnement net entrant

$$R_n = H + LE + G + S$$

C'est la différence entre le rayonnement solaire entrant et le rayonnement thermique sortant. Le rayonnement solaire est absorbé par la canopée et converti en chaleur.

$$Rn = (1 - \alpha) * R_s - \varepsilon * R_l$$

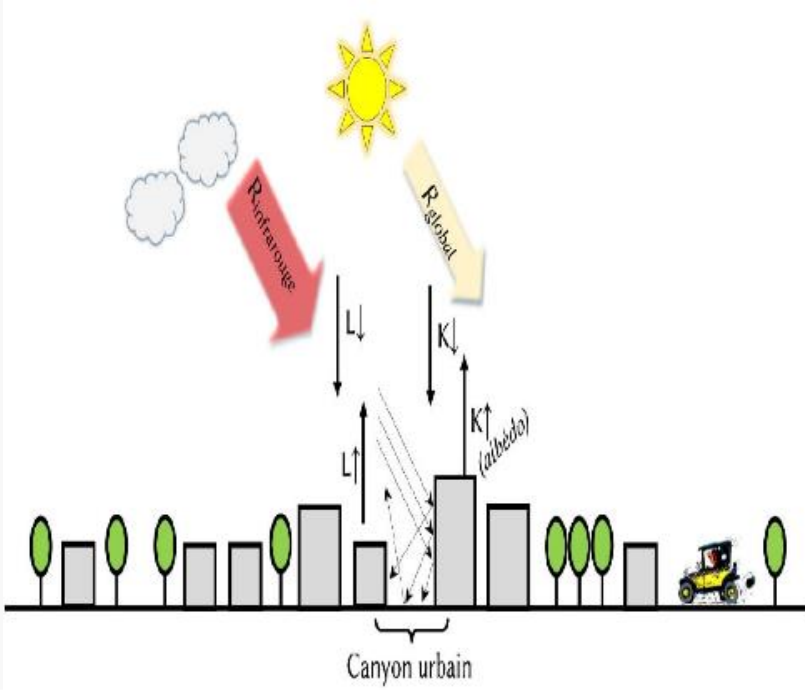
Avec:

Albedo (α) : L'albédo est la proportion de rayonnement solaire réfléchi par une surface.

Rayonnement solaire entrant (R_s) : Le rayonnement solaire entrant est la quantité d'énergie solaire qui atteint la surface terrestre.

Emissivité (ε) : L'émissivité est une mesure de la capacité d'une surface à émettre du rayonnement thermique.

R_l Rayonnement thermique sortant : Le rayonnement thermique sortant est la quantité de chaleur émise par les surfaces des bâtiments et la surface du sol dans l'atmosphère.



$$R_l = \sigma * T_s^4$$ où

ε :est l'émissivité de la surface

σ : est la constante de Stefan-Boltzmann,

Ts: est la température de surface de l'élément considéré.

Flux de chaleur sensible

$$R_n = H + LE + G + S$$

Le flux de chaleur sensible est le flux de chaleur qui résulte de la différence de température entre l'air et les surfaces des bâtiments.

$$H = \rho * Cp * (Ta - Ts) * U$$

Avec:

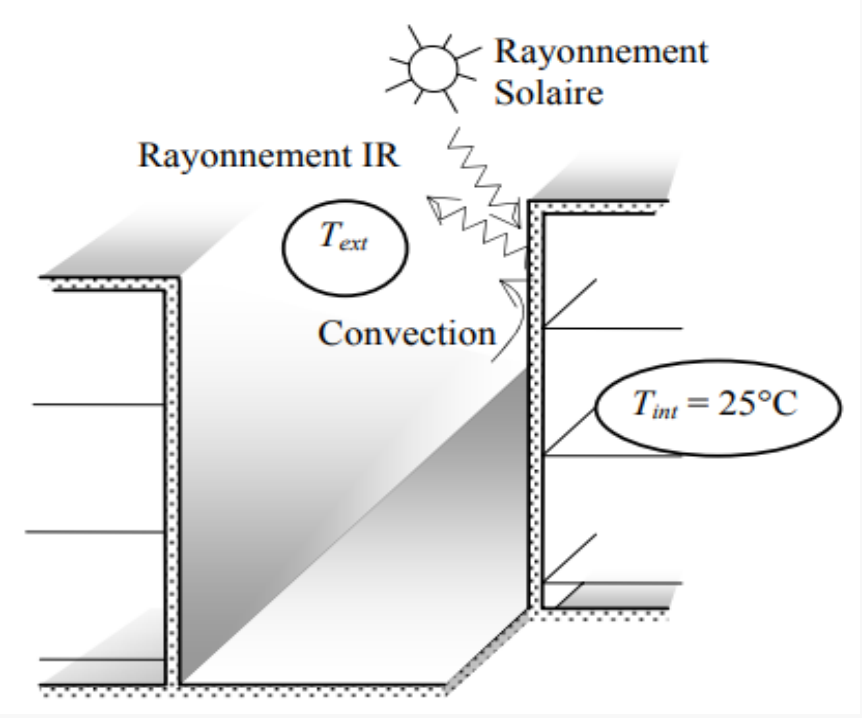
ρ :est la densité de l'air

Cp :est la capacité thermique spécifique de l'air

Ta :est la température de l'air

Ts :est la température de surface des bâtiments

U :est la vitesse du vent.



Flux de chaleur latente

$$R_n = H + LE + G + S$$

Le flux de chaleur latente est le flux de chaleur qui résulte de l'évaporation de l'eau à la surface des bâtiments et de la transpiration des plantes dans le canyon urbain.

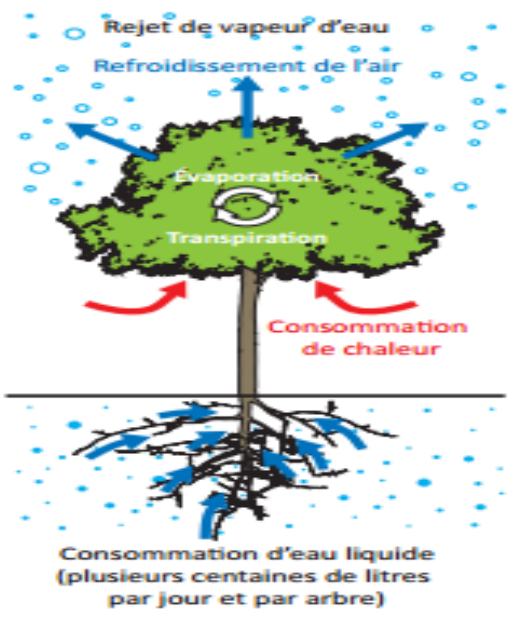
$$LE = \rho * L * U * (qs - qa)$$

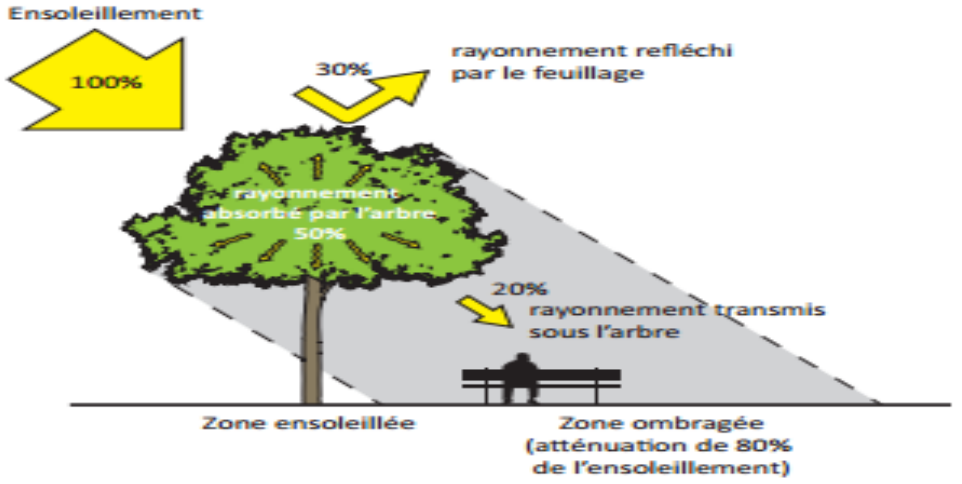
Avec :

L :est la chaleur latente de vaporisation de l'eau

qs :est la pression partielle de vapeur d'eau à la surface des bâtiments

qa :est la pression partielle de vapeur d'eau dans l'air.





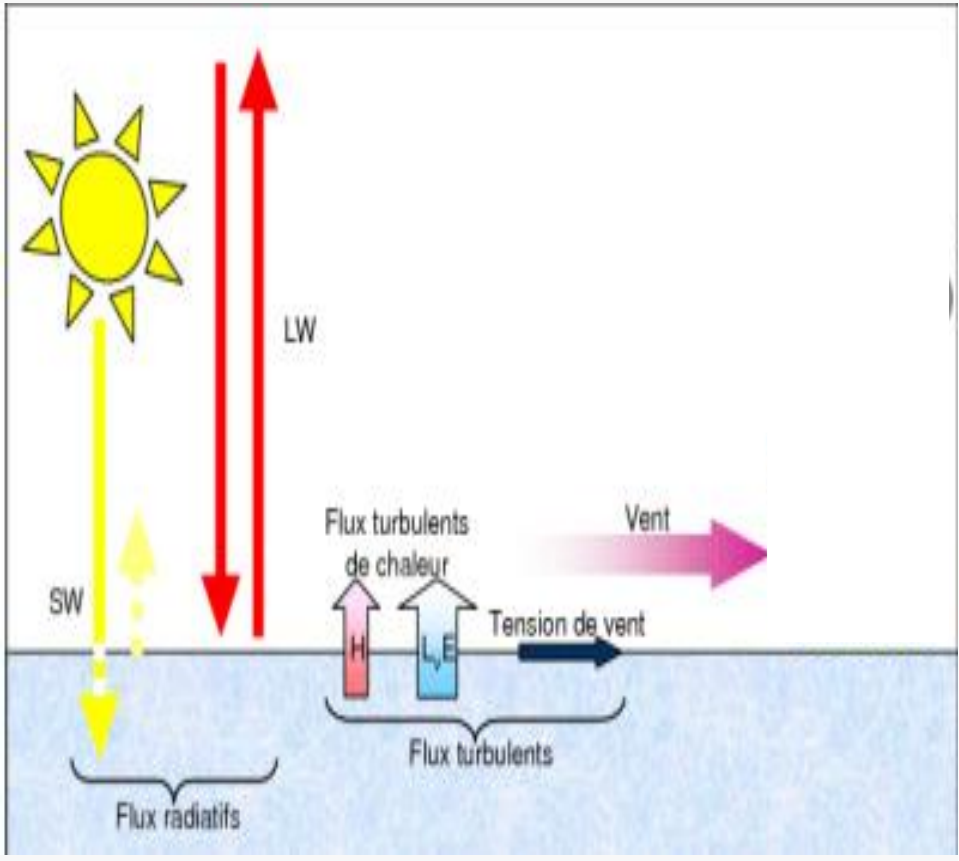
Flux de chaleur du sol

$$R_n = H + LE + G + S$$

Le flux de chaleur du sol est le flux de chaleur qui résulte de l'échange de chaleur entre la surface du sol et l'atmosphère.
Il est calculé en utilisant la formule :

$$G = \lambda * (Ts - Tg)$$

Avec:
λ :est la conductivité thermique du sol
Ts :est la température de surface du sol
Tg :est la température du sol à une profondeur donnée.



Stockage de chaleur

$$R_n = H + LE +$$

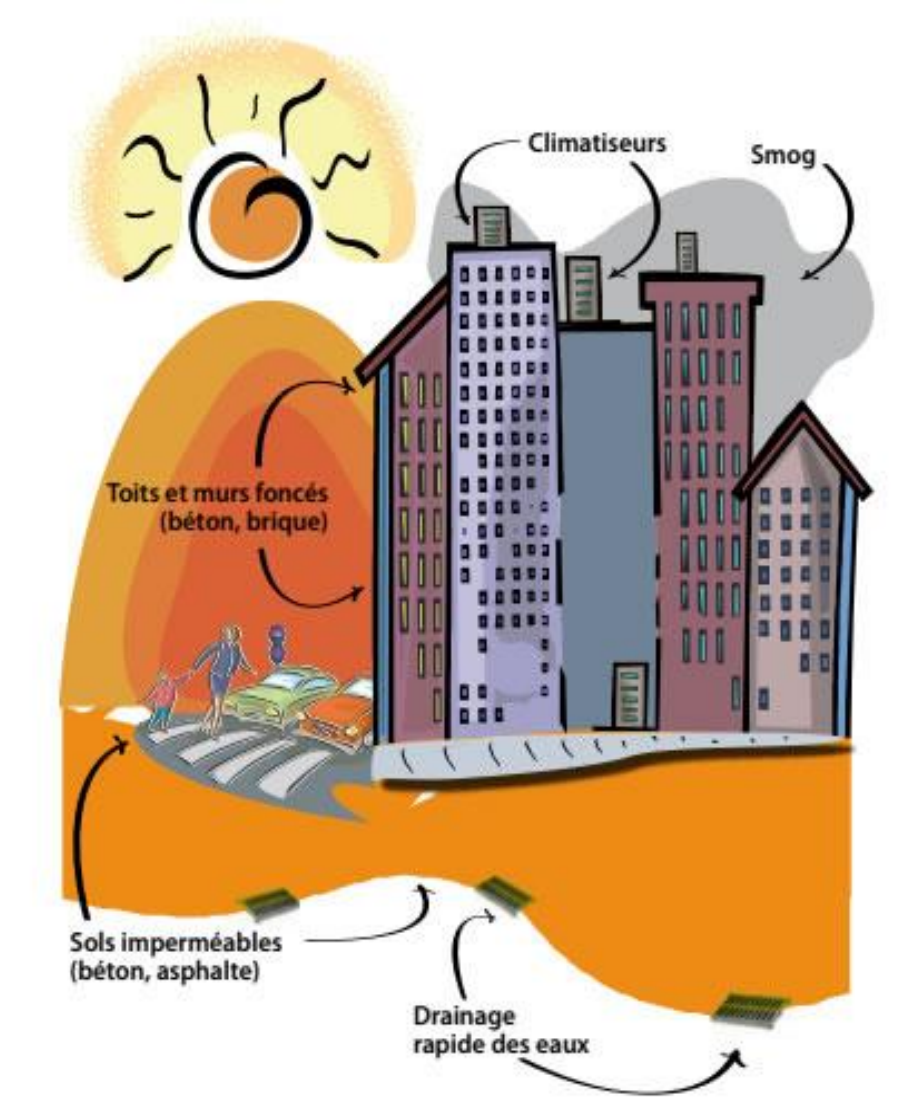
Le stockage de chaleur est la quantité de chaleur stockée dans les matériaux de surface des bâtiments, tels que les murs, les toits et les routes.

$$S = \rho * Cs * \frac{dTs}{dt}$$ Avec:

ρ : Densité la densité est la masse volumique du matériau en kg/m^3.

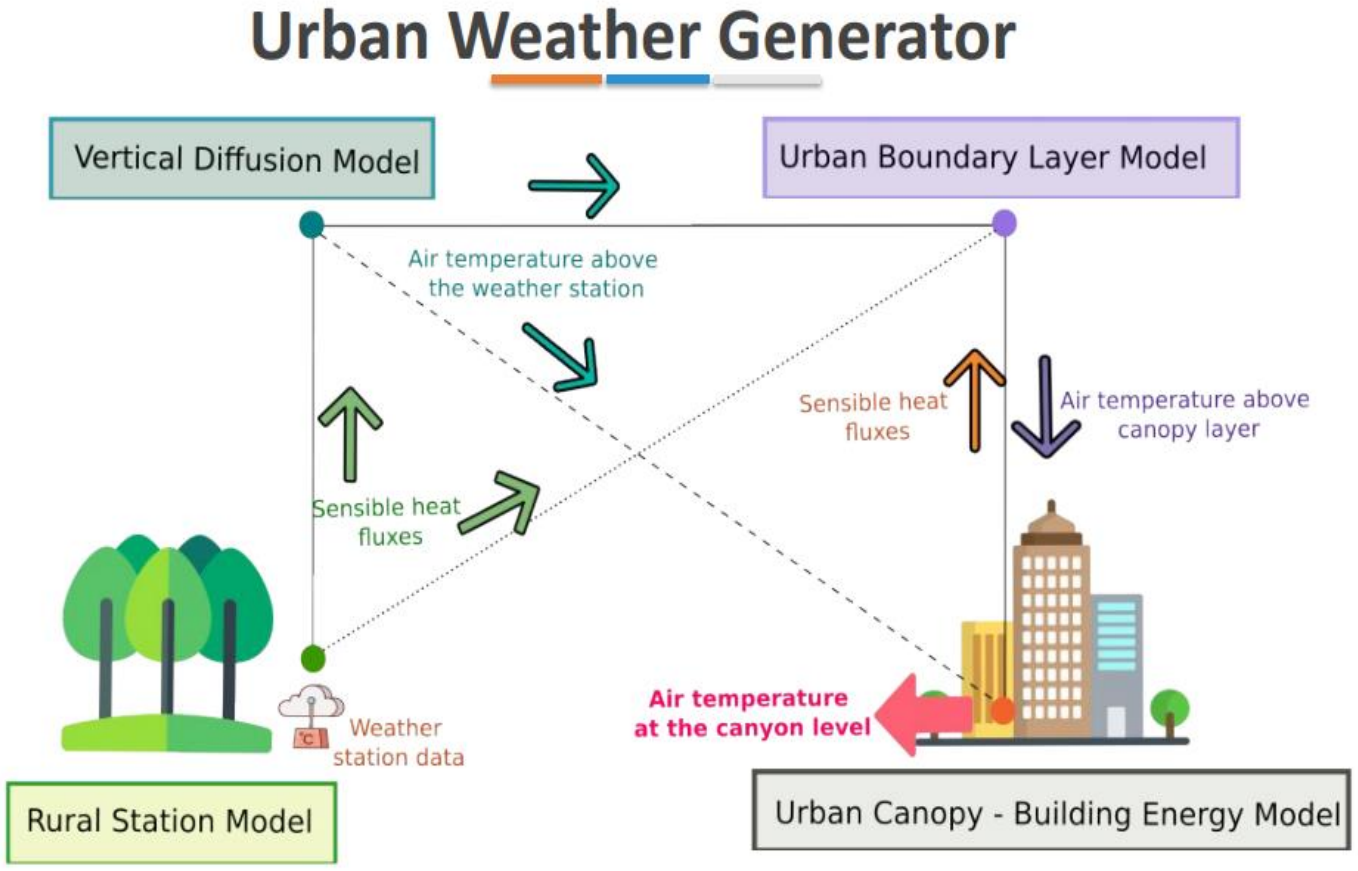
Cs :Capacité thermique massique J/kg.K.

$\frac{dTs}{dt}$:C'est un indicateur de la vitesse à laquelle la chaleur est stockée ou libérée par les matériaux de surface.



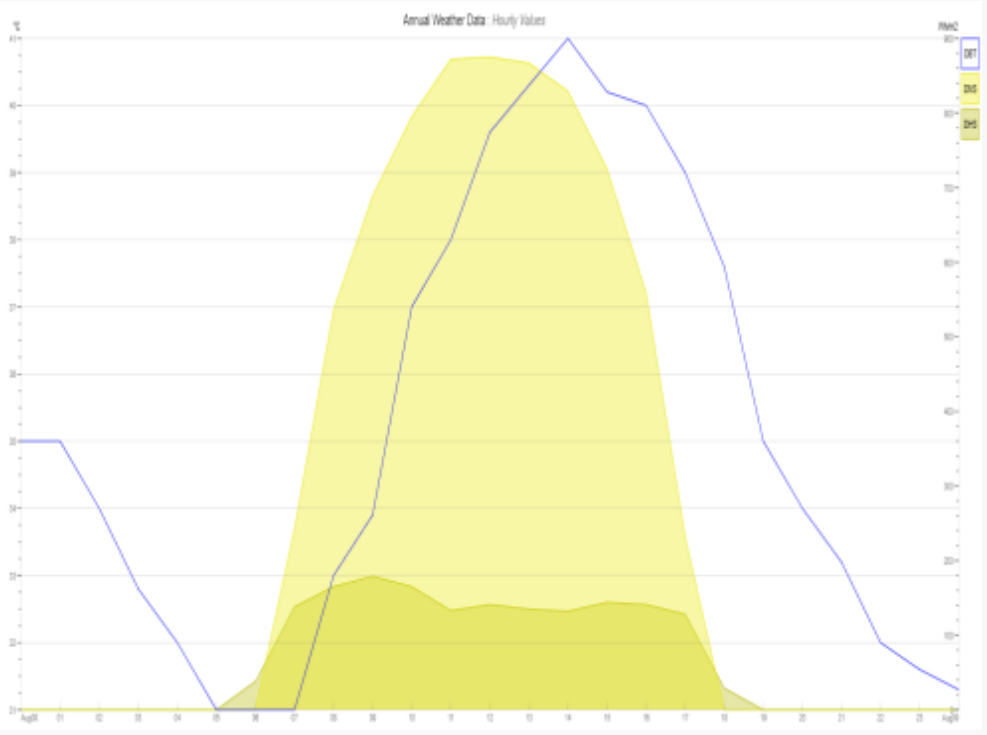
3. Implémentation du modèle en python et résultats

Urban Weather Generator



Cas étudié : Masdar Abu Dhabi

Masdar Abu Dhabi



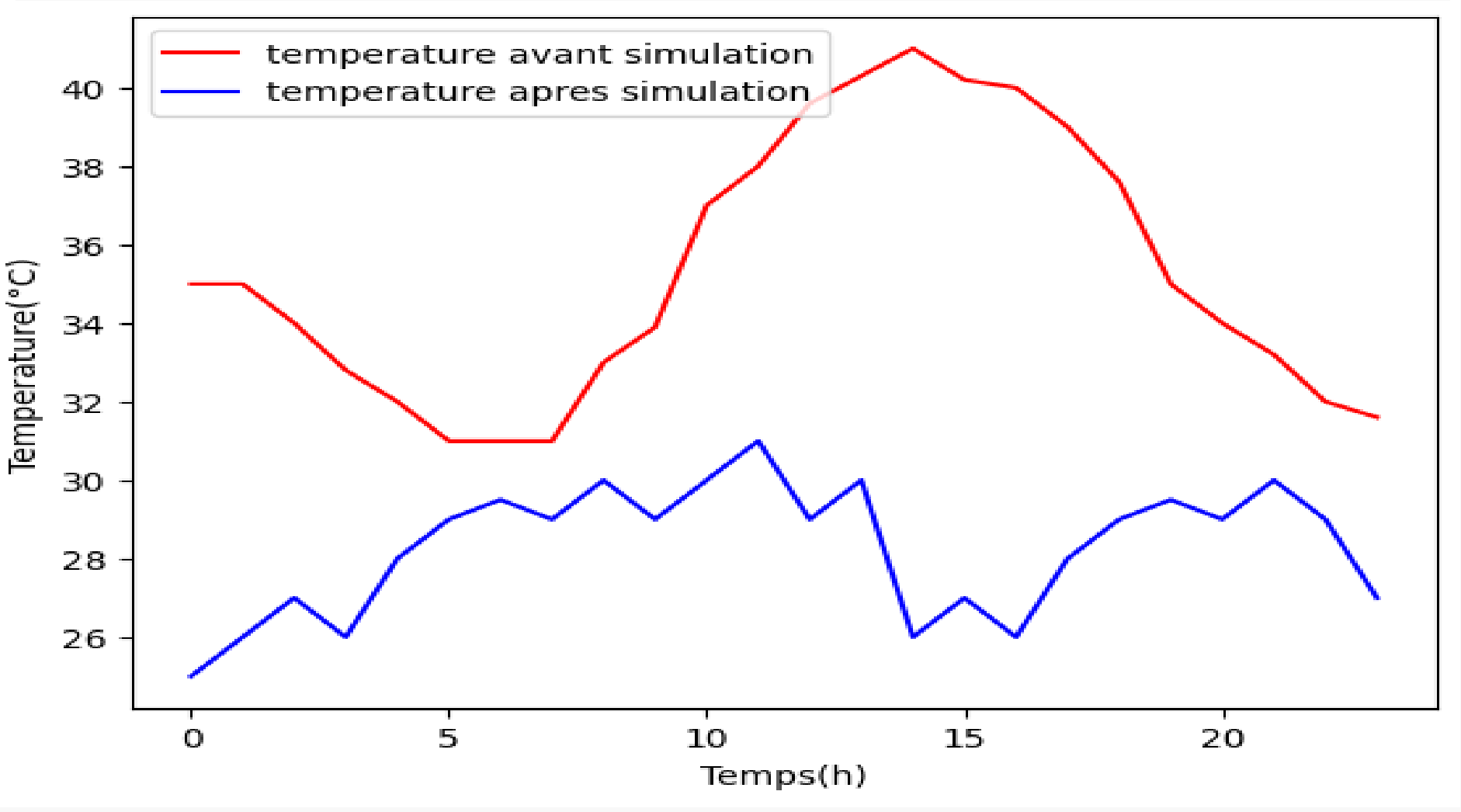
Température et rayonnement solaire le 8 aout 1997

Fichier epw

```
<epw.epw.epw object at 0x7f85f37ea890>
{'LOCATION': ['ABU DHABI', '-', 'ARE', 'IWEC Data', '412170', '24.43', '54.65', '4.0', '27.0'],
   Year  Month  Day  Hour  Minute  \
0  1997      1    1     1      60
1  1997      1    1     2      60
2  1997      1    1     3      60
3  1997      1    1     4      60
4  1997      1    1     5      60

              Data Source and Uncertainty Flags  Dry Bulb Temperature  \
0  C9C9C9C9*0?9?9?9?9?9?9?9A7A7C8C8A7*0*0E8*0*0                  16.0
1  C9C9C9C9*0?9?9?9?9?9?9?9A7A7C8C8A7*0*0E8*0*0                  15.6
2  C9C9C9C9*0?9?9?9?9?9?9?9A7A7C8C8A7*0*0E8*0*0                  15.1
3  C9C9C9C9*0?9?9?9?9?9?9?9A7A7C8C8A7*0*0E8*0*0*0                14.8
4  C9C9C9C9*0?9?9?9?9?9?9?9A7A7A7C8A7A7*0E8*0*0                  14.4

   Dew Point Temperature  Relative Humidity  Atmospheric Station Pressure  \
0                   12.3                 78                        101700
1                   12.5                 82                        101700
2                   12.6                 85                        101600
3                   12.6                 87                        101600
4                   12.5                 88                        101600

   ...  Ceiling Height  Present Weather Observation  Present Weather Codes  \
0  ...           77777                            9              999999999
1  ...           77777                            9              999999999
2  ...           77777                            9              999999999
3  ...           77777                            9              999999999
4  ...           22000                            9              999999999

   Precipitable Water  Aerosol Optical Depth  Snow Depth  \
0                   0                  0.055           0
1                   0                  0.055           0
2                   0                  0.055           0
3                   0                  0.055           0
4                   0                  0.055           0
```

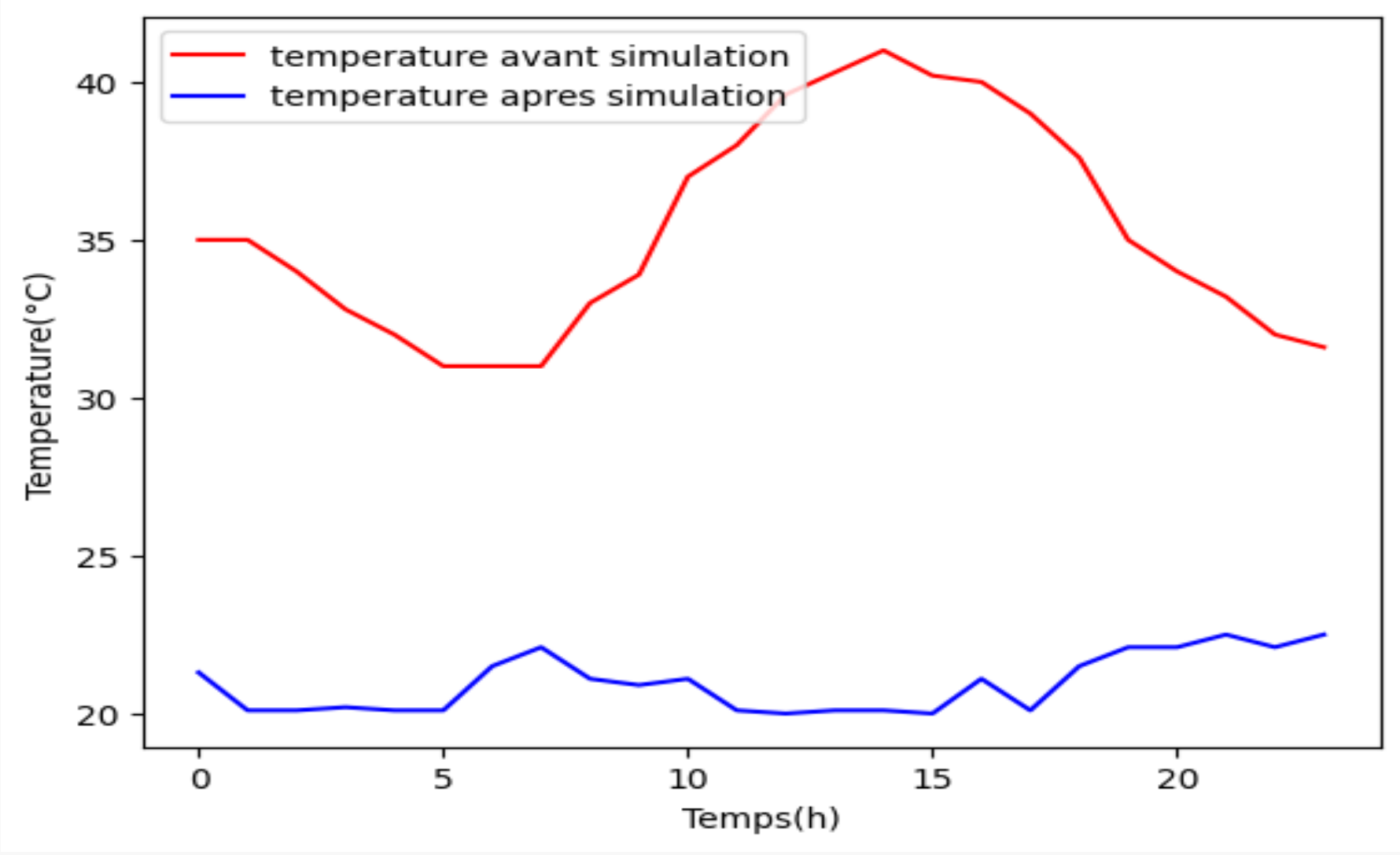
Première simulation avec des paramètres aléatoires

```
#Initialiser le modèle UWG en s'appuyant sur les valeurs par défaut
model = UWG.from_param_args(epw_path=epw_path, bldheight=20, blddensity=0.5,
                            grasscover=0.1, treecover=0.5)
```



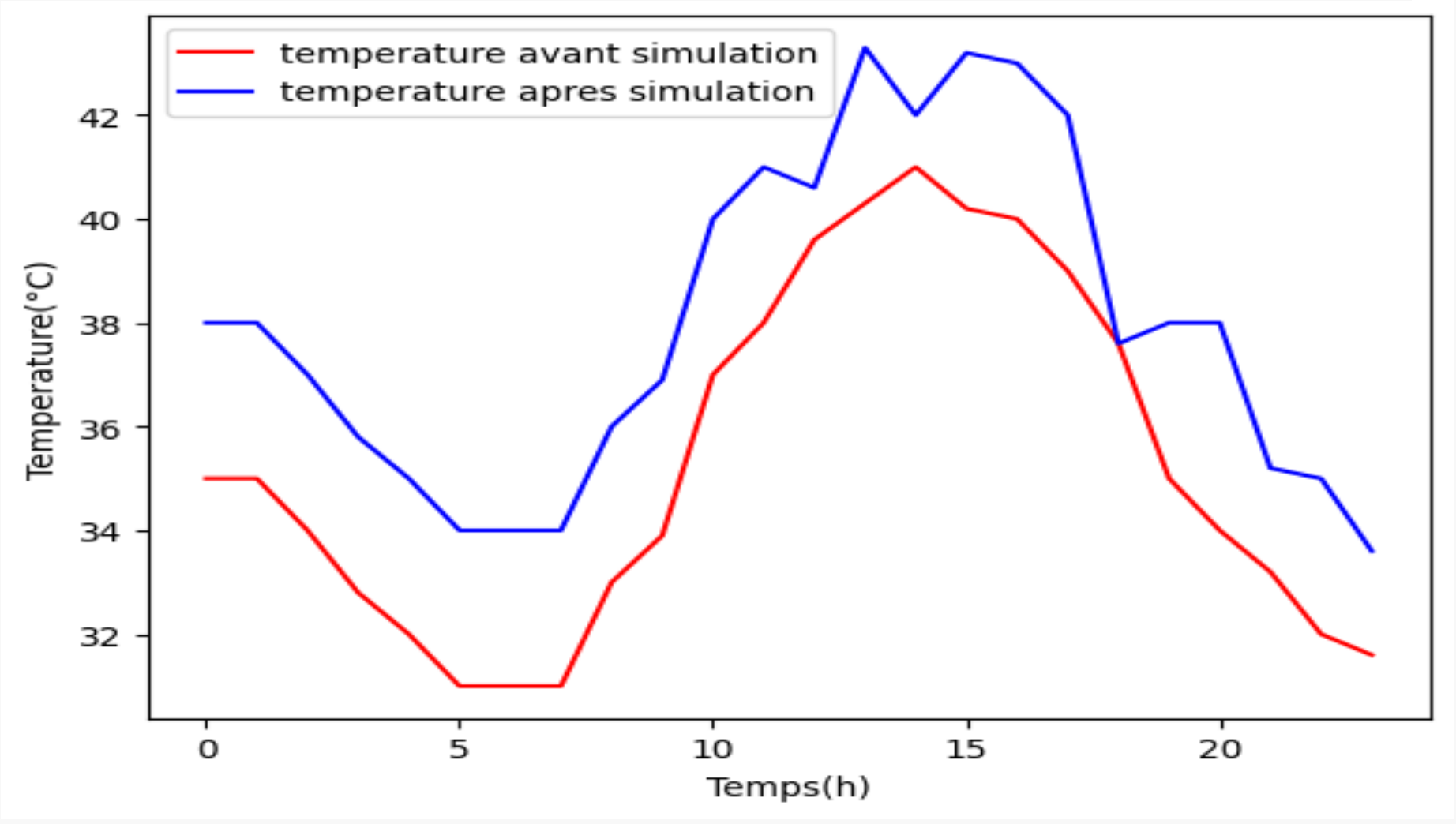
Deuxième simulation en changeant les paramètres de végétation : augmentation de la densité de végétation

```
# Initialiser le modèle UWG en passant des paramètres comme arguments
model = UWG.from_param_args(epw_path=epw_path, bldheight=20, blddensity=0.5,
                            grasscover=0.1, treecover=0.9)
```



Troisième simulation en changeant les paramètres des bâtiments: augmentation de la densité des bâtiments

```
# Initialiser le modèle UWG en passant des paramètres comme arguments
model = UWG.from_param_args(epw_path=epw_path, bldheight=150, blddensity=0.9,
                            grasscover=0.1, treecover=0.2)
```



Conclusion

Merci pour votre attention

Annexe

- Les Scripts Python
  - Importer les bibliothèques nécessaires
  - Visualiser quelques données
  - Simulation
  - Traçage de courbes

Importer les bibliothèques nécessaires

```
# bibliothèques pour le modèle urban weather generator
pip install uwg
import uwg
cd uwg
!pip install -r dev-requirements.txt
!pip install -r requirements.txt

# bibliothèques pour lire les fichiers epw
pip install git+https://github.com/building-
energy/epw.git@master
Import epw
# bibliothèques pour tracer les courbes
import matplotlib.pyplot as plt
```

Visualiser quelques données

```
14 from epw import epw
15 a=epw()
16 a.read(r'/content/ARE_Abu.Dhabi.412170_IWEC.epw')
17 print(a)
18 df=a.dataframe  # this is pandas dataframe
19 f1=df[['Year', 'Month', 'Day', 'Hour', 'Minute','Dry
20 Bulb Temperature']]
21 b=df.loc[df['Year']==1993]
22 p=b.loc[b['Month']==8]
23 o=p.loc[p['Day']==8]
24 print(o)
```

Visualiser quelques données

```
<epw.epw.epw object at 0x7f85f0de19f0>
      Year  Month  Day  Hour  Minute  \
5256  1993      8    8     1      60
5257  1993      8    8     2      60
5258  1993      8    8     3      60
5259  1993      8    8     4      60
5260  1993      8    8     5      60
5261  1993      8    8     6      60
5262  1993      8    8     7      60
5263  1993      8    8     8      60
5264  1993      8    8     9      60
5265  1993      8    8    10      60
5266  1993      8    8    11      60
5267  1993      8    8    12      60
5268  1993      8    8    13      60
5269  1993      8    8    14      60
5270  1993      8    8    15      60
5271  1993      8    8    16      60
5272  1993      8    8    17      60
5273  1993      8    8    18      60
5274  1993      8    8    19      60
5275  1993      8    8    20      60
5276  1993      8    8    21      60
5277  1993      8    8    22      60
5278  1993      8    8    23      60
5279  1993      8    8    24      60

                 Data Source and Uncertainty Flags  Dry Bulb Temperature
5256  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8C8A7*0*0E8*0*0                  35.0
5257  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8C8A7*0*0E8*0*0                  35.0
5258  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8C8A7*0*0E8*0*0                  34.0
5259  A7A7E8E8*0?9?9?9?9?9?9?9A7A7E8C8A7*0*0E8*0*0                  32.8
5260  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8C8A7*0*0E8*0*0                  32.0
5261  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8C8A7*0*0E8*0*0                  31.0
5262  A7A7E8E8*0H9H9H9I9I9I9I9A7A7E8A7A7A7*0E8*0*0                  31.0
5263  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7*0*0E8*0*0                  31.0
5264  A7A7E8B8*0H9H9H9I9I9I9I9A7A7E8B8A7A7*0E8*0*0                  33.0
5265  A7A7E8E8*0G9G9G9I9I9I9I9A7A7A7A7A7A7*0E8*0*0                  33.9
5266  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7A7*0E8*0*0                  37.0
5267  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7*0*0E8*0*0                  38.0
5268  A7A7E8E8*0G9G9G9I9I9I9I9A7A7A7A7A7A7*0E8*0*0                  39.6
5269  B8C8E8B8*0H9H9H9I9I9I9I9*0B8B8B8B8*0*0E8*0*0                  40.3
5270  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7*0*0E8*0*0                  41.0
5271  A7A7E8E8*0G9G9G9I9I9I9I9A7A7A7A7A7A7*0E8*0*0                  40.2
5272  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7*0*0E8*0*0                  40.0
5273  A7A7E8B8*0H9H9H9I9I9I9I9A7A7B8B8A7*0*0E8*0*0                  39.0
5274  A7A7E8E8*0G9G9G9I9I9I9I9A7A7A7A7A7A7*0E8*0*0                  37.6
5275  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8B8A7*0*0E8*0*0                  35.0
5276  A7A7E8B8*0?9?9?9?9?9?9?9A7A7B8B8A7*0*0E8*0*0                  34.0
5277  A7A7E8E8*0?9?9?9?9?9?9?9A7A7A7A7A7A7*0E8*0*0                  33.2
```

Simulation

```
25 from uwg import uwg
26
27 # première simulation
28
29 # Définissez les chemins .epw, .uwg pour créer un objet uwg.
30 epw_path = "/content/ARE_Abu.Dhabi.412170_IWEC.epw" #
31 available in resources directory.
32
33 # Initialiser le modèle UWG en s'appuyant sur les valeurs
34 par défaut ou aléatoires
35 model = UWG.from_param_args(epw_path=epw_path,
36 bldheight=20, blddensity=0.5,
37                   , grasscover=0.1, treecover=0.5)
38 # simulation
39 model.generate()
40 model.simulate()
41 # Écrire le résultat de la simulation dans un fichier.
42 model.write_epw()
43
```

Simulation

```
44 from uwg import uwg
45
46 # deuxième simulation
47
48 # Définissez les chemins .epw, .uwg pour créer un objet uwg.
49 epw_path = "/content/ARE_Abu.Dhabi.412170_IWEC.epw"
50
51 # Initialiser le modèle UWG en s'appuyant sur les valeurs
52 par défaut ou aléatoires
53 model = UWG.from_param_args(epw_path=epw_path,
54 bldheight=20, blddensity=0.5,
55                     , grasscover=0.1, treecover=0.9)
56 # simulation
57 model.generate()
58 model.simulate()
59 # Écrire le résultat de la simulation dans un fichier.
60 model.write_epw()
61
62
```

Simulation

```
63 from uwg import uwg
64
65 # troisième simulation
66
67 # Définissez les chemins .epw, .uwg pour créer un objet uwg.
68 epw_path = "/content/ARE_Abu.Dhabi.412170_IWEC.epw"
69
70 # Initialiser le modèle UWG en s'appuyant sur les valeurs
71 par défaut ou aléatoires
72 model = UWG.from_param_args(epw_path=epw_path,
73 bldheight=150, blddensity=0.9,
74                   , grasscover=0.1, treecover=0.2)
75 # simulation
76 model.generate()
77 model.simulate()
78 # Écrire le résultat de la simulation dans un fichier.
79 model.write_epw()
80
81
```

Traçage de courbes

```
82  import matplotlib.pyplot as plt
83
84  # Données de temps et de température
85  temps = [0, 1, 2, 3, 4,
86  5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
87  Temperature1=[35.0, 35.0, 34.0, 32.8, 32.0, 31.0,
88  31.0, 31.0, 33.0, 33.9, 37.0, 38.0, 39.6, 40.3, 41.0,
89  40.2, 40.0, 39.0, 37.6, 35.0, 34.0, 33.2, 32.0, 31.6]
90  Temperature2 =[38.0, 38.0, 37.0, 35.8, 35.0, 34.0,
91  34.0, 34.0, 36.0, 36.9, 40.0, 41.0, 40.6, 43.3, 42.0,
92  43.2, 43.0, 42.0, 37.6, 38.0, 38.0, 35.2, 35.0, 33.6]
93  # Tracer la courbe
94  plt.plot(temps,Temperature1,color='red',label="tempera
95  ture avant simulation")
96  plt.plot(temps,Temperature2,color='blue',label="temper
97  ature apres simulation")
98  # Ajouter des labels aux axes
99  plt.xlabel('Temps(h)')
100 plt.ylabel('Temperature(°C)')
101 plt.legend()
102 # Afficher la courbe
103 plt.show()
```