

# Modélisation Thermique du Bâtiment Fauriel

Sébastien Natchez

IMT – MINES Saint-Étienne – Institut Henri Fayol – LIMOS UMR 6158

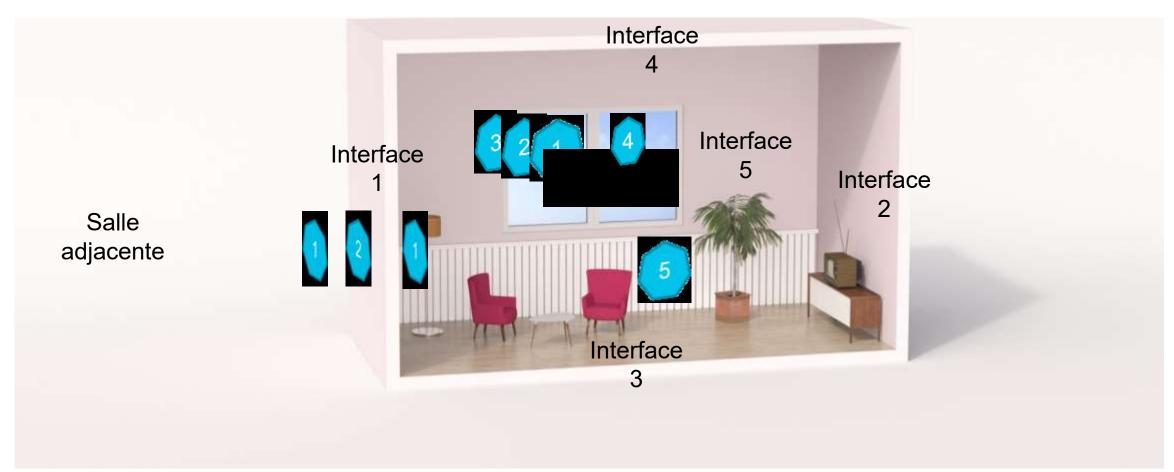




### Les objectifs à réaliser

- 1. Comprendre les phénomènes thermiques mis en jeu
  - 2. Connaître les caractéristiques du bâtiment
    - 3. Choisir la méthode de modélisation

## Une vue du problème : phénomènes thermiques dans une salle



1 : Convection Naturelle entre l'air de la salle et la surface de l'interface

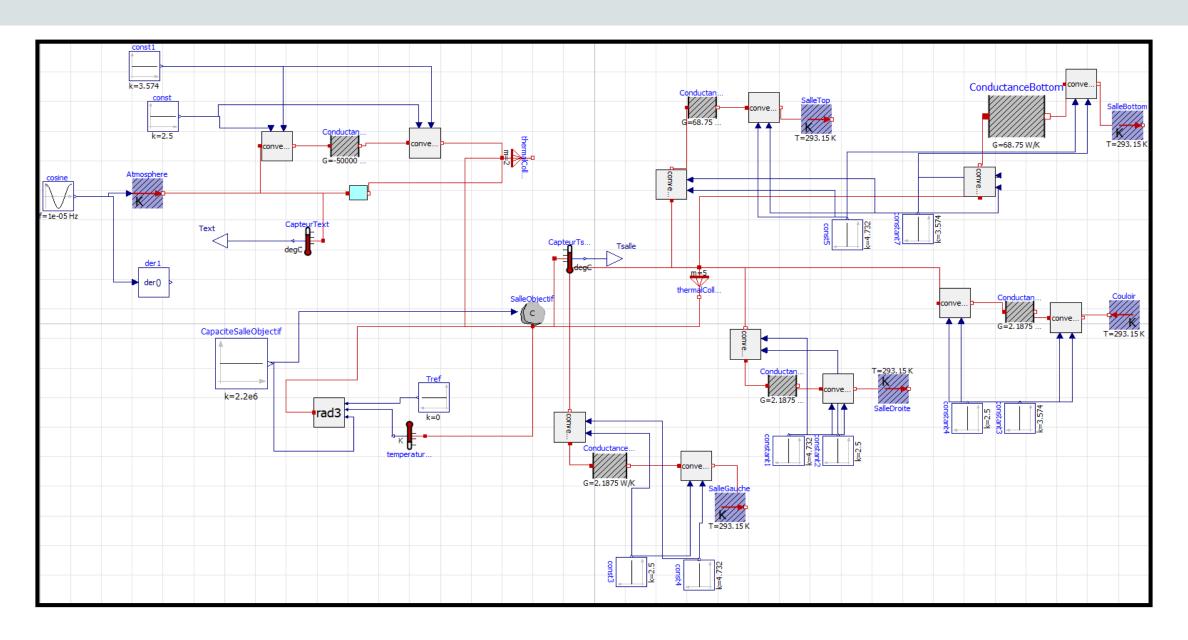
2 : Conduction dans l'interface

3 : Convection forcée entre l'air de l'atmosphère et la surface de l'interface

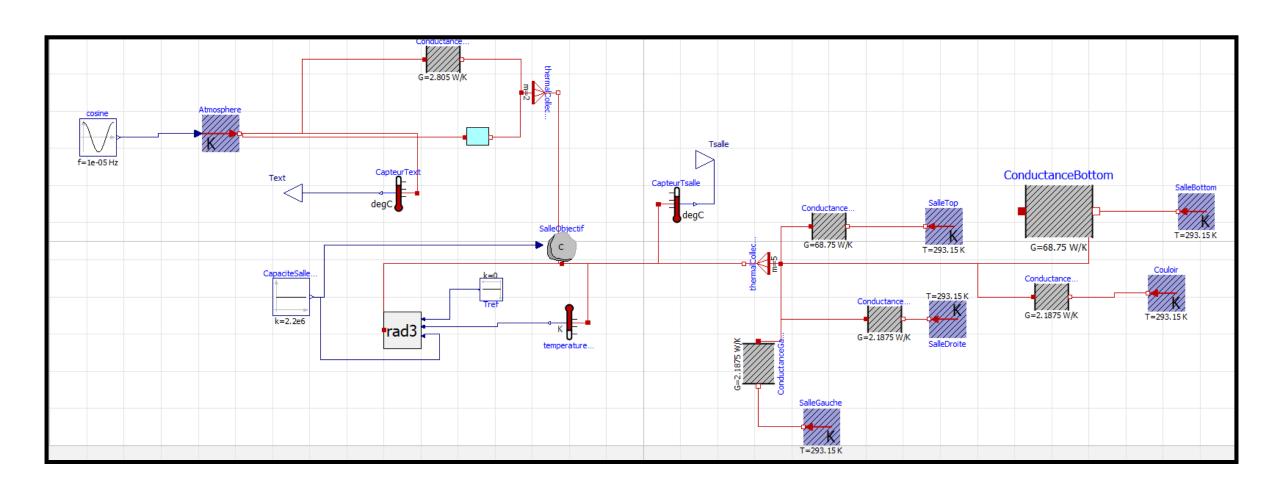
4 : Apports thermiques solaires

5 : Radiateur

### Modélisation d'une salle : Les deux modèles



### Modélisation d'une salle : Les deux modèles



## Détermination des conductances/inertie d'une salle : La méthode

- →On exporte la modélisation OpenModelica en fichier FMU
- →On recupère les données des capteurs thermiques des salles intéressantes
- → Avec Python on insère les données de température dans le modèle exporté en FMU
- →On utilise l'algorithme de Nelder-Mead pour trouver les valeurs de conductances et d'inertie thermique.

### Détermination des conductances/inertie d'une salle :

### Le code

```
capteursPath="C:\\Users\\sebastien.natchez\\Desktop\\Modelisation multi physique Fauriel\\Python\\Python FMU\\"
        capteurFilename="log.2022-0*.csv"
        knxPath="C:\\Users\\sebastien.natchez\\Desktop\\Modelisation_multi_physique_Fauriel\\Python\\Python FMU\\"
        knxFilename="knx metrics 20220*.csv"
       metricsPath="C:\\Users\\sebastien.natchez\\Desktop\\Modelisation_multi_physique_Fauriel\\Python\\Python\FMU\\"
       metricsFilename="metrics 20220*.csv"
        (arrayCapteurs1, arrayknx1, arrayMetrics1)=Get Arrays(capteursPath,capteurFilename,knxPath,knxFilename,metricsPath,metricsFilename)
        tinit= 1649028600.0-2*3600-86400-22*3600
        tfini=1649028600.0-2*3600
        dateinit=datetime.datetime.fromtimestamp(tinit)
        datefini=datetime.datetime.fromtimestamp(tfini)
        (temps, synchroCapteur, synchroknx, synchroMetrics)=Get Synchrone Donnees From Arrays(arrayCapteurs1,arrayknx1,arrayMetrics1,tinit,tfini)
110
        (Tobjectif, Temperature) = Recupere Temperature From Synchrone(synchroCapteur, synchroknx, synchroMetrics)
        #On dit que le béton est une sorte de thermostat donc la température évolue comme une moyenne à chaue instant
        #de la température des 2 salles l'entourent
       MoyenneT325=0
115
116
       MoyenneT529=0
       for i in range(len(temps)):
117
           MoyenneT325+=(Temperature[3][i]+Tobjectif[i])/2/len(temps)
           MoyenneT529+=(Temperature[2][i]+Tobjectif[i])/2/len(temps)
119
       for i in range(len(temps)):
120
           Temperature[3][i]=(Temperature[3][i]+Tobjectif[i])/2
            Temperature[2][i]=(Temperature[2][i]+Tobjectif[i])/2
        #On passe en Kelvin pour la simulation et on met le départ à 0 pour la simulation
123
       t0=temps[0]
124
       for i in range(len(Temperature)):
125
           NM.Degre To Kelvin(Temperature[i])
126
       NM.Degre To Kelvin(Tobjectif)
127
       for i in range(len(temps)):
128
           temps[i]-=t0
129
           temps[i]=temps[i]/3600
        (T425, Tcoincafe, T529, T325plafond, Tcouloir, Text) = (Temperature [0], Temperature [1], Temperature [2], Temperature [3], Temperature [4], Temperature [5])
       T429=Tobjectif
```

### Détermination des conductances/inertie d'une salle :

Le code

```
fmu filename = 'SalleSimple.fmu'
# ------
# fmu filename = 'Salle Complete.fmu'
def Simule(inertieConductances):
   return NM.run experiment(fmu filename, temps, Temperature ,Tobjectif, inertieConductances)
def Simule Compare(inertieConductances):
   courbeSimulee=Simule(inertieConductances)
   distance=NM.Calcul Distance(courbeSimulee,Tobjectif)
   return distance
tupleInitial=[51000,4.375,4.375,45.375,45.375,4.375,3.3,11.52]
#bnds=((1e4,1e9),(0.5,20),(0.5,20),(1,20),(1,20),(0.5,20),(0.5,20),(0.5,20))
bnds=((1e4,8e4),(1,1e9),(1,1e9),(1,1e9),(1,1e9),(1,1e9),(1,1e9),(1,1e9))
result = minimize(Simule Compare, tupleInitial, method='nelder-mead',bounds=bnds, options={"maxiter":50})
newTab=result['x']
Tsimule=Simule(newTab)
print(newTab)
print(NM.Calcul_Distance(Tobjectif, Tsimule))
print(dateinit)
print(datefini)
plot(temps, Tsimule, 'r', temps, Tobjectif, 'b')
for i in range(3):
   result = minimize(Simule Compare, tupleInitial, method='nelder-mead',bounds=bnds, options={"maxiter":50}}
   newTab=result['x']
Tsimule=Simule(newTab)
print(newTab)
print(NM.Calcul Distance(Tobjectif, Tsimule))
print(dateinit)
print(datefini)
plot(temps, Tsimule, 'r', temps, Tobjectif, 'b')
```

135

### Détermination des conductances/inertie d'une salle :

### Le code

```
#On calcule l'équivalent conductivité des valeurs trouvées
       equivalentConductivite=deepcopy(newTab)
179
       h=2.5
       LmurFen=3.574
       Lmur=4.732
       hfen=1.601
       Lfen=1.653
       equivalentConductivite[1]=equivalentConductivite[1]*0.07/(h*Lmur)
       equivalentConductivite[2]=equivalentConductivite[2]*0.07/(h*Lmur)
       equivalentConductivite[3]=equivalentConductivite[3]*0.2/(LmurFen*Lmur)
       equivalentConductivite[4]=equivalentConductivite[4]*0.2/(LmurFen*Lmur)
       equivalentConductivite[5]=equivalentConductivite[5]*0.07/(h*LmurFen)
       equivalentConductivite[6]=equivalentConductivite[6]*0.58/(h*LmurFen)
       equivalentConductivite[7]=equivalentConductivite[7]*0.2/(hfen*Lfen)
       print(equivalentConductivite)
```

```
TabInitial=[51000,4.375,4.375,45.375,45.375,4.375,3.3,11.52]
       pourcentageSouhaite=1
217
       Tsimule=Simule(TabInitial)
218
       erreur0 = NM.Calcul Distance(T429, Tsimule)
219
       erreurRelative0=(erreur0-0)/erreur0*100
       meilleurTableau=deepcopy(TabInitial)
221
       erreurPrecedente=erreur0
222
       #erreurVariable=erreurPrecedente*(1-pourcentageSouhaite-1/1000)
       erreurVariable=erreur0
224
       while True:
           for i in range(len(meilleurTableau)):
                tableauTransition=deepcopy(meilleurTableau)
               tableauTransition[i] *= 2
               Tsimule=Simule(tableauTransition)
               if NM.Calcul Distance(T429, Tsimule) <=erreurVariable:</pre>
                    erreurVariable=NM.Calcul Distance(T429, Tsimule)
                    #print('erreur calculee', erreurVariable)
                    tableauVariable=tableauTransition
                   # print((erreurPrecedente-erreurVariable)/erreurPrecedente*100)
           for i in range(len(meilleurTableau)):
               tableauTransition=deepcopy(meilleurTableau)
               tableauTransition[i] /= 2
               Tsimule=Simule(tableauTransition)
               print('-erreur', NM.Calcul Distance(T429, Tsimule))
               if NM.Calcul Distance(T429, Tsimule) <=erreurVariable:</pre>
                    erreurVariable=NM.Calcul Distance(T429, Tsimule)
240
                    #print('erreur calculee', erreurVariable)
                    tableauVariable=tableauTransition
                    #print((erreurPrecedente-erreurVariable)/erreurPrecedente*100)
           print('meilleur amelioration relative ',(erreurPrecedente-erreurVariable)/erreurPrecedente*100)
           print('son 8-uplets est ',tableauVariable)
           if (erreurPrecedente-erreurVariable)/erreurPrecedente*100>pourcentageSouhaite:
               erreurPrecedente=erreurVariable
               meilleurTableau=tableauVariable
           else:
                print("plus d'amelioration significative possible du tuple : ", meilleurTableau )
               TsimuleMeilleur=Simule(meilleurTableau)
               print("l'erreur est désormais de ", NM.Calcul Distance(T429, TsimuleMeilleur))
               break
       plot(temps, Tsimule, 'r', temps, Tobjectif, 'b')
```

# Détermination des conductances/inertie d'une salle de la salle 429

- →On souhaite appliquer la méthode à la salle 429 (salle de Maxime)
- →On place des capteurs autour de cette salle (425, couloir, plafond 325...)
- →On coupe le chauffage, ferme la porte et la fenêtre et on utilise les données lorsque la salle est inoccupée.
- →Les fenêtres laissent passer la lumière, néanmoins l'orientation de la fenêtre ne permet pas la captation direct de rayon du soleil. On négligera donc l'apport thermique solaire.

# Détermination des conductances/inertie d'une salle de la salle 429 : Modélisation sur un week-end

#### Résultats attendus :

→ Capacité thermique élevée : 7-8\*10^6

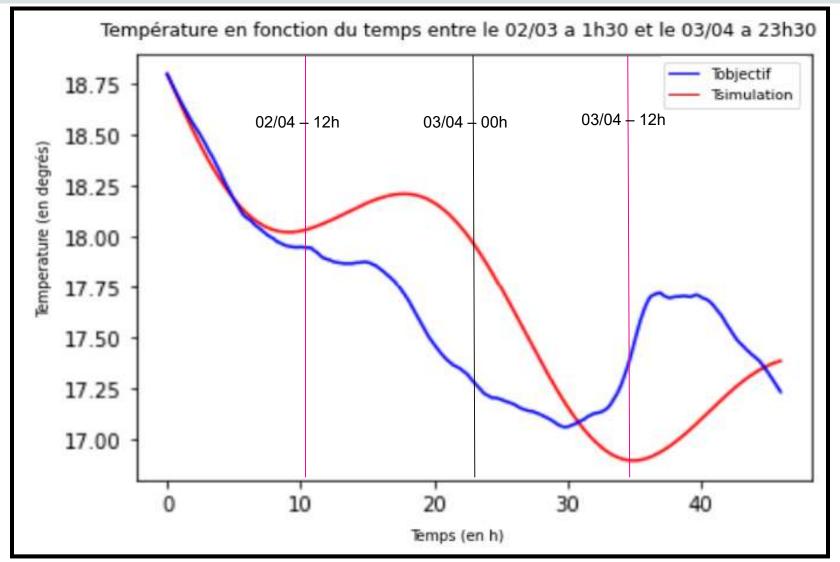
→ Symétrie haut/bas, droite/gauche

→ Conductance de la fenetre proportionnellement très élevée

→ Valeurs proches de l'audit :

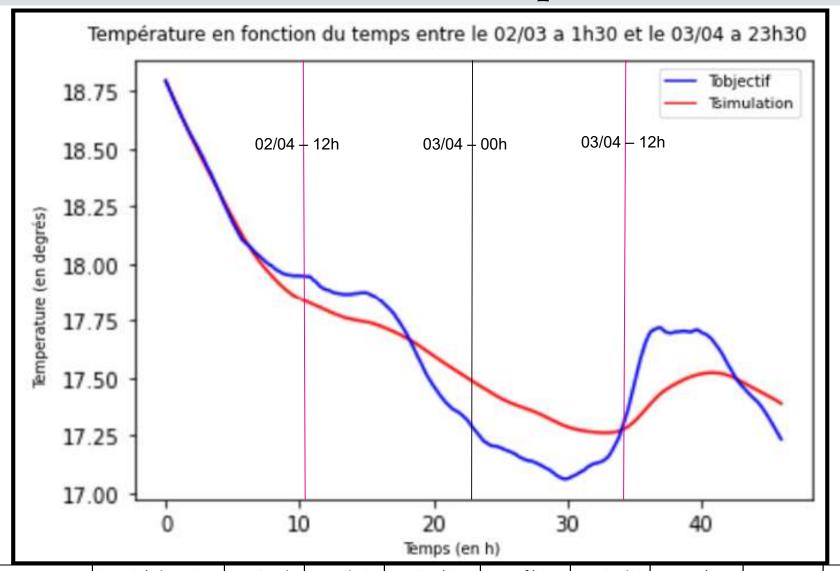
	Mur Gauche	Mur Droit	Plancher supérieur	Plancher inférieur	Mur Couloir	Mur Fenêtré	Fenêtre
Surface	11.83	11.83	16.912168	16.912168	8.935	8.935	2.646453
Épaisseur	0.07	0.07	0.2	0.2	0.07	0.58	0.2
Conductance	5.915	5.915	32.5325	32.5325	4.4675	2.94855	7.62178464
Conductivité thermique	0.035	0.035	0.55	0.55	0.035	0.1914	0.576
Matériaux	placo-platre	placo-platre	Béton plein	Béton plein	placo-platre	Calcaire + PSE	Double-vitrage

# Détermination des conductances sur un week-end avec le modèle complet



	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Avec convection	8.67E+05	1.05	1.47	992.31	2.37	1.02	5.30	4.62	0.30
Résultat attendu	7-8E+06	5.92	5.92	32.53	32.53	4.47	2.95	7.62	

# Détermination des conductances sur un week-end avec le modèle simple



									4
	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Conduction pure	2.33E+06	12.13	32.29	71.27	1.04	7.73	1.00	1.17	0.12
Résultat attendu	7-8E+06	5.92	5.92	32.53	32.53	4.47	2.95	7.62	

### Conclusion

					Conductances	5			
	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Avec convection	8.67E+05	1.05	1.47	992.31	2.37	1.02	5.30	4.62	0.30
Conduction pure	2.33E+06	12.13	32.29	71.27	1.04	7.73	1.00	1.17	0.12
Résultat attendu	7-8E+06	5.92	5.92	32.53	32.53	4.47	2.95	7.62	

				Conductivités th	nermiques				
	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Avec convection	8.67E+05	0.01	0.01	11.73	0.03	0.01	0.34	0.35	0.30
Conduction pure	2.33E+06	0.07	0.19	0.84	0.01	0.06	0.06	0.09	0.12
Résultat attendu	7-8E+06	0.035	0.035	0.55	0.55	0.035	0.1914	0.58	

- → La modélisation la plus simple permet d'obtenir de meilleures courbes
- → Les valeurs sont fausses

# Détermination des conductances/inertie d'une salle de la salle 429 : Modélisation sur différentes nuits

#### Salle avec les modèles de convection

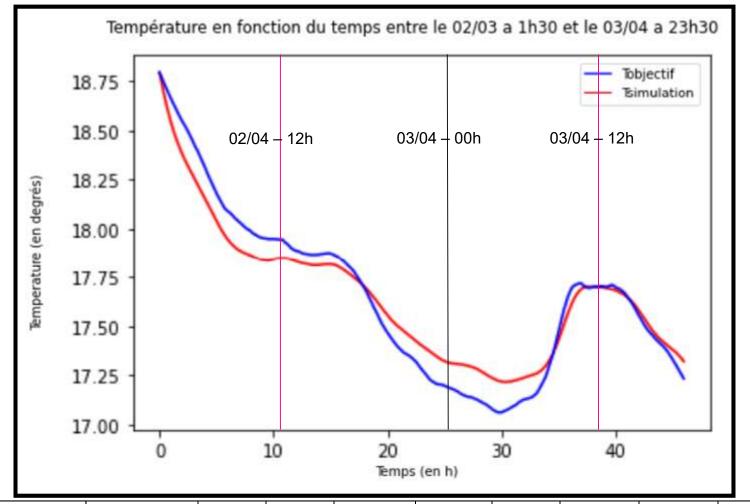
				Matin									Soir				
Date	Capacité Thermique	Xavier	Coin Cafe	529	325	Couloir	Mur Fenetre	Fenetre	Date	Capacité Thermique	Xavier	Coin Cafe	529	325	Couloir	Mur Fenetre	Fenetre
24/03/2022									24/03/2022	1.36E+06	3.36	1.57	26.65	3.80	9.98	2.41	3.30
25/03/2022	1.44E+06	4.58	1.54	24.55	33.88	2.91	3.82	4.87	25/03/2022	7.23E+05	12.79	3.98	8.13	165.53	61.64	1.06	3.87
26/03/2022	1.24E+06	1.58	3.98	15.88	414.10	6.94	1.58	2.95	26/03/2022	1.79E+06	4.60	1.03	94.03	4.23	1.33	1.52	2.36
27/03/2022	1.38E+06	1.24	4.60	17.02	68.84	5.50	1.56	2.76	27/03/2022	1.42E+06	3.01	5.00	5.03	225.49	4.22	1.44	1.81
28/03/2022	2.74E+06	1.95	1.67	31.27	6.05	1.74	2.80	3.60	28/03/2022	1.86E+06	2.90	1.14	111.60	45.08	1.00	1.04	3.45
29/03/2022	1.53E+06	2.60	4.83	46.87	30.87	6.37	2.50	1.33	29/03/2022								
30/03/2022	2.48E+06	3.56	1.17	81.07	4.48	6.90	1.49	2.12	30/03/2022		•		•				

#### Salle en conduction pure

Į ,				Matin									Soir				
Date	Capacité Thermique	Xavier	Coin Cafe	529	325	Couloir	Mur Fenetre	Fenetre	Date	Capacité Thermique	Xavier	Coin Cafe	529	325	Couloir	Mur Fenetre	Fenetre
24/03/2022			I						24/03/2022	1.96E+06	4.37	2.12	9.04	1.00	11.71	4.19	2.61
25/03/2022	3.16E+06	1.08	2.05	18.39	19.54	3.19	1.29	8.67	25/03/2022	2.43E+06	1.38	3.46	30.85	145.25	4.62	1.07	1.06
26/03/2022	2.47E+06	2.19	3.37	1.28	39.63	2.95	3.09	3.55	26/03/2022	8.33E+06	10.37	2.80	113.68	147.96	1.07	1.06	1.00
27/03/2022	2.26E+06	1.28	2.53	1.04	129.61	4.35	3.08	2.30	27/03/2022	2.18E+06	4.66	3.73	1.00	77.87	2.58	1.03	1.00
28/03/2022	5.71E+06	1.40	1.00	1.18	18.70	1.03	4.20	6.47	28/03/2022	2.24E+06	35.21	24.01	109.97	1.11	2.35	1.04	1.02
29/03/2022	4.47E+06	1.08	6.14	1.34	39.63	3.66	4.50	4.39	29/03/2022								
30/03/2022	2.89E+06	3.77	6.75	1.02	1.00	5.30	1.99	2.15	30/03/2022								
																-	

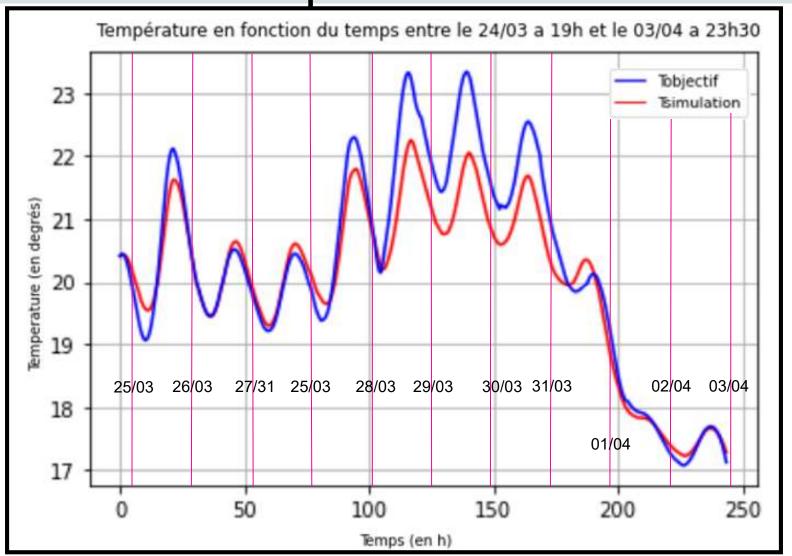
→ On observe des tendances pour certaines valeurs mais elles ne correspondent à rien et ne sont pas cohérentes entres-elles

# Détermination des conductances/inertie d'une salle de la salle 429 : Modification du modèle



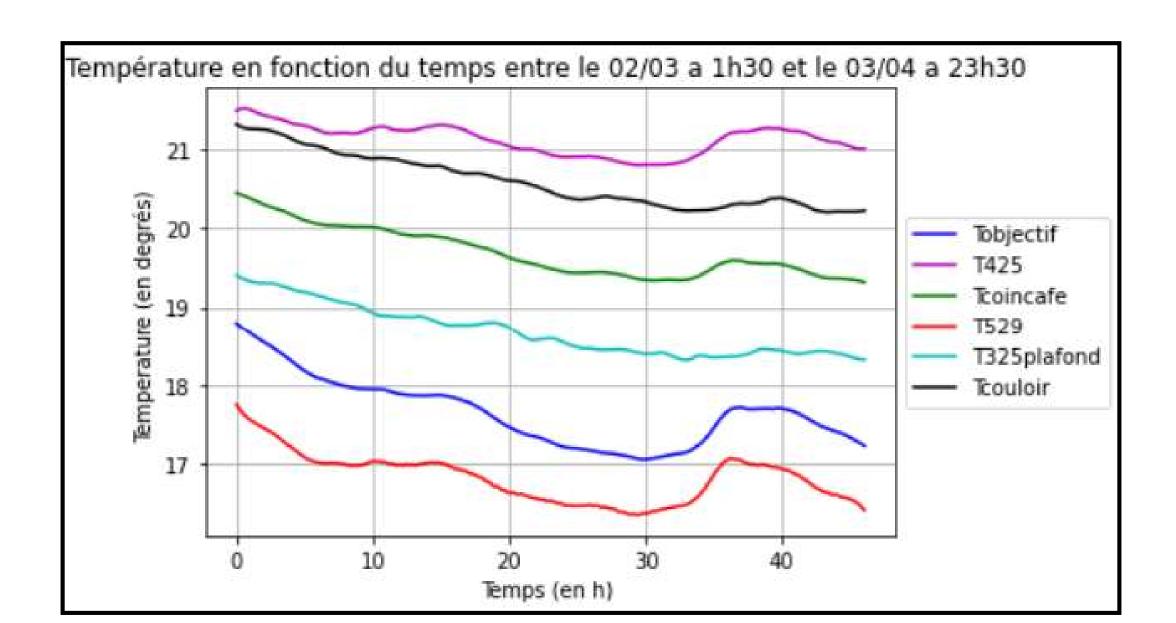
	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Conductances	1.66E+06	1.37	80.78	316.65	1.24	37.83	1.13	1.09	0.09
Conductivités	1.66E+06	0.01	0.48	3.74	0.01	0.30	0.07	0.08	0.09
Résultat attendu :	7-8E+06	5.92	5.92	32.53	32.53	4.47	2.95	7.62	
Conductances									
Résultat attendu :	7-8E+06	0.035	0.035	0.55	0.55	0.035	0.1914	0.58	
Conductivités	7-0L+00	0.033	0.033	0.55	0.55	0.033	0.1514	0.56	

Simulation d'une longue période avec le résultat précédent

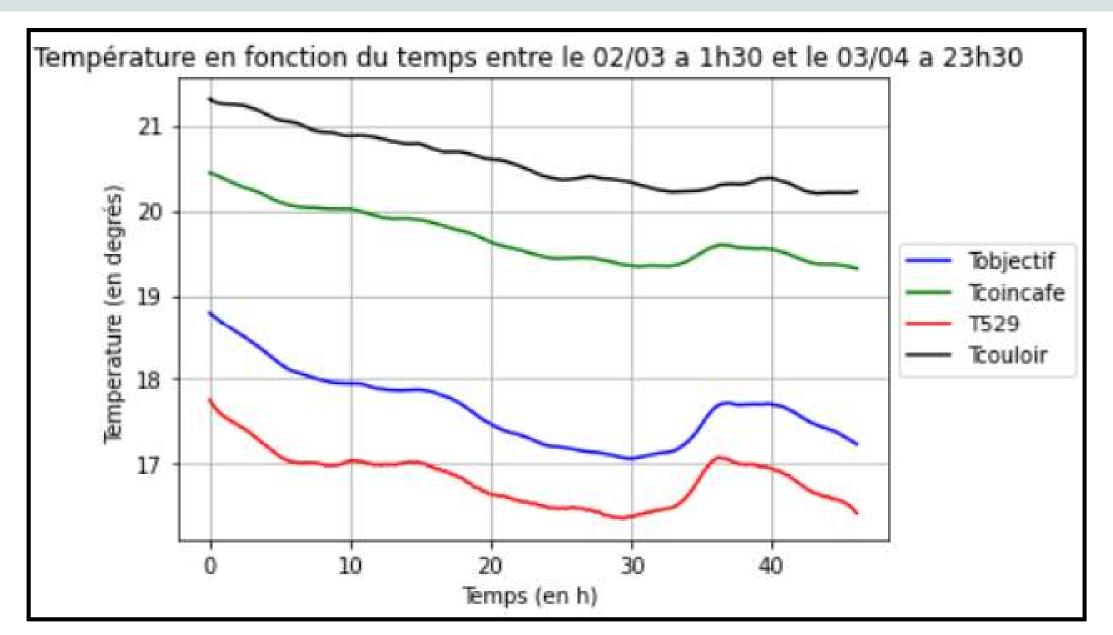


	Capacité Thermique	Mur Gauche	Mur droit	Mur Supérieur	Mur Inférieur	Mur Couloir	Mur extérieur	Fenetre	Erreur
Conduction pure	1.66E+06	1.37	80.78	316.65	1.24	37.83	1.13	1.09	0.37
Résultat attendu	7-8E+06	5.92	5.92	32.53	32.53	4.47	2.95	7.62	

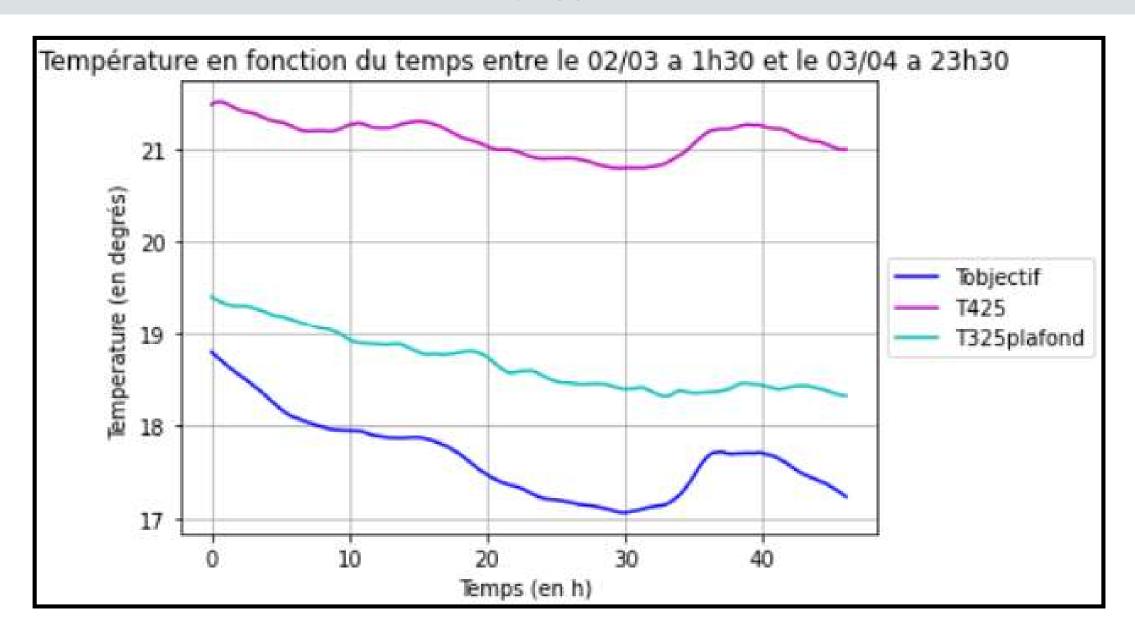
### Températures extérieures



# Températures correspondantes aux conductances élevées



## Températures correspondantes aux conductances faibles



### Conclusion

### Les sources d'erreurs

- → Précision des capteurs (précision pure et erreur en journée)
- →Dépendance des températures des salles entres-elles : Les évolutions sont identiques
- → Variation de température dans une salle trop faible
- → Modèle