

Régulation d'un bâtiment thermiquement actif

Introduction aux méthodes numériques et projet
Première année de bachelier ingénieur civil, Université de Liège
Année académique 2024-2025

1 Introduction

L'importance croissante de l'efficacité énergétique dans la construction a conduit à développer de nouveaux systèmes de climatisation de bâtiments, en particulier dans le secteur tertiaire (Figure 1). Dans ce cadre, le chauffage et refroidissement par des éléments en béton où des tubes remplis d'eau sont intégrés est une technique moderne permettant de maintenir le confort thermique tout en restant efficace d'un point de vue énergétique (Figure 2). Les dalles de béton servent alors de système de stockage d'énergie thermique pour chauffer ou refroidir. Un des avantages de ce système, appelé TABS ("Thermo-Active Building Systems"), est que les composants sont pratiquement sans entretien et offrent une totale liberté pour une conception flexible de l'espace.

L'application des TABS demande cependant aux ingénieurs de comprendre et de contrôler la dynamique thermique du bâtiment afin de pouvoir implémenter cette solution. Cette technologie peut par exemple être couplée à un système de pompe à chaleur et de puits géothermiques afin d'obtenir un système efficace et durable de chauffage et de refroidissement.

Ce projet a pour objectif le développement d'un modèle simplifié de bâtiment TABS couplé à un système géothermique permettant de réguler la température intérieure et d'atteindre un confort thermique optimal tout en consommant moins d'énergie primaire.

2 Description du modèle

La dynamique thermique du bâtiment est modélisée par un système simplifié (Figure 3) dont les variables sont les températures et les paramètres, les résistances thermiques, les capacités thermiques spécifiques et les flux de chaleur entrants ou sortants.

L'air de la pièce est supposé être parfaitement mélangé de telle sorte que l'on puisse le traiter comme un élément ayant une température uniforme. On considère que la capacité thermique de la dalle en béton est divisée en 3 : la partie supérieure, la partie centrale, et la partie inférieure. Les autres capacités thermiques incluent celle de l'eau et celle des tubes. Le contact entre la dalle de béton et l'air de la pièce est idéalisé par une surface.



FIGURE 1 – Les bureaux du Centre of Sustainable Building à l’Université de Cassel (Allemagne) sont dotés d’un système d’activation thermique de noyaux de béton (voir Figure 2).

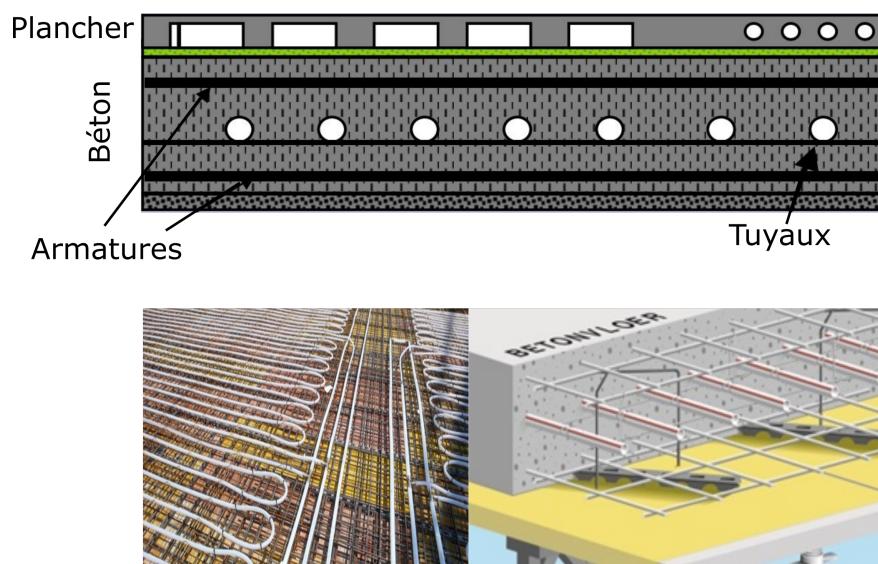


FIGURE 2 – Système TABS dans une dalle en béton.

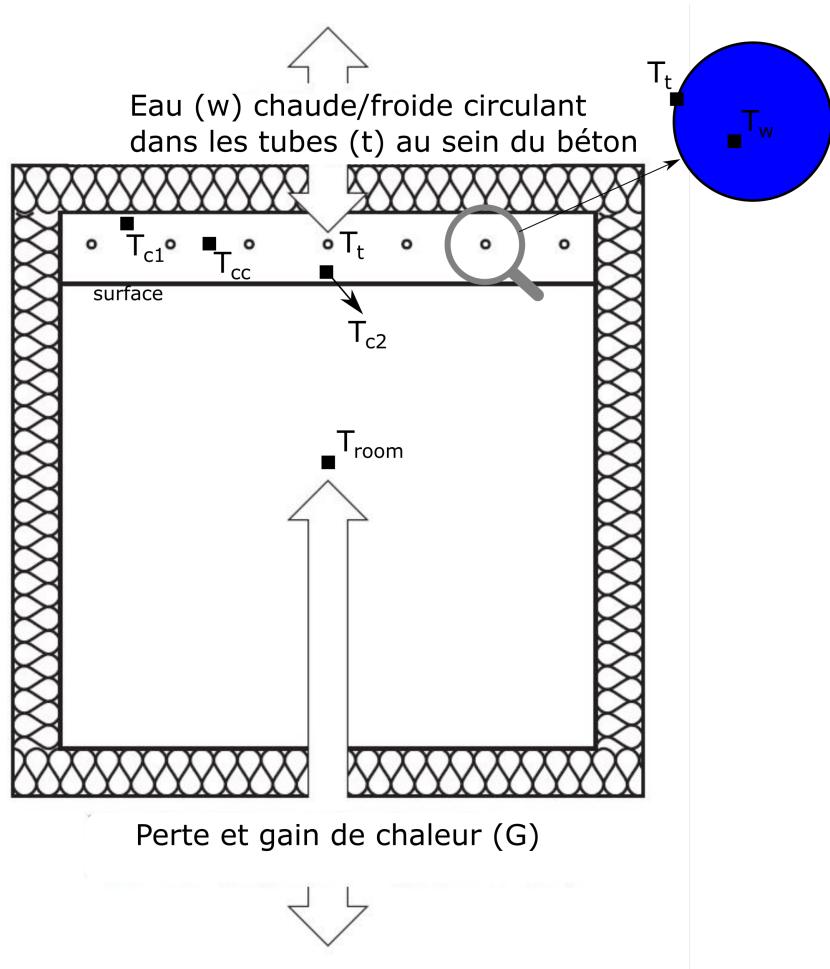


FIGURE 3 – Modèle simplifié du bâtiment (coupe verticale).

Le bilan énergétique pour le modèle de la Figure 3 s'écrit en fonction des différences de température entre chaque élément considéré et les éléments adjacents comme suit.

La partie centrale du béton est en contact avec les parties supérieure et inférieure ainsi qu'avec les tubes :

$$C_{cc} \frac{dT_{cc}}{dt} = -\frac{1}{R_{cc-c1}}(T_{cc} - T_{c1}) - \frac{1}{R_x}(T_{cc} - T_t) + \frac{1}{R_{c2-cc}}(T_{c2} - T_{cc}) \quad (1)$$

La partie supérieure du béton est en contact avec la partie centrale du béton :

$$C_{c1} \frac{dT_{c1}}{dt} = -\frac{1}{R_{cc-c1}}(T_{c1} - T_{cc}) \quad (2)$$

La partie inférieure du béton est en contact avec la pièce régulée à travers la surface et avec la partie centrale du béton :

$$C_{c2} \frac{dT_{c2}}{dt} = -\frac{1}{R_{c2-cc}}(T_{c2} - T_{cc}) + \frac{1}{R_{r-s} + R_{s-c2}}(T_{room} - T_{c2}) \quad (3)$$

La pièce régulée subit des pertes et des gains et est en contact avec la partie inférieure du béton à travers la surface :

$$C_{room} \frac{dT_{room}}{dt} = -\frac{1}{R_{r-s} + R_{s-c2}}(T_{room} - T_{c2}) + G(t) \quad (4)$$

avec

- T : les températures en K ;
- dT/dt : les dérivées des températures en K/s ;
- R : les résistances thermiques en Km^2/W ;
- C : les capacités thermiques spécifiques en $\text{J}/(\text{Km}^2)$;
- $G(t)$: les pertes et gains de chaleur en fonction du temps t en W/m^2 ;

et où les indices représentent

- t , les tubes ;
- $c1$, la partie supérieure du béton ;
- cc , la partie centrale du béton ;
- $c2$, la partie inférieure du béton ;
- s , la surface entre le béton et la pièce régulée ;
- $room$, la pièce régulée ;
- x le contact entre les tubes et la partie centrale du béton.

Les valeurs numériques des paramètres intervenant dans le problème sont reprises dans la Table 3 (attention aux unités).

3 Chauffage et refroidissement

Le TABS régule le système à travers l'eau parcourant les tubes intégrés dans la dalle en béton. On considère que les tubes sont en contact avec la partie centrale du béton et

l'eau. Le bilan énergétique pour les tubes s'écrit suivant :

$$C_w \frac{dT_t}{dt} = -\frac{1}{R_x}(T_t - T_{cc}) - \frac{1}{R_w}(T_t - T_w) \quad (5)$$

où l'indice w représente l'eau au sein des tubes. En mode chauffage, l'eau provient d'une pompe à chaleur à 28°C, on a donc $T_w = 28^\circ\text{C}$. En mode refroidissement, les tubes sont connectés directement à un échangeur de chaleur relié au puits géothermique à 18°C, on a donc $T_w = 18^\circ\text{C}$. Lorsque le chauffage et le refroidissement sont mis à l'arrêt, on considère que le dernier terme de l'équation (5) est annulé. Le système de pilotage doit être conçu de telle sorte que l'on puisse (1) chauffer/refroidir/mettre à l'arrêt en fonction d'un programme de 24h donné ou (2) démarrer le chauffage/refroidissement/mise à l'arrêt en fonction de certaines variables, par exemple une température de confort.

Question 1 : Recherche de racines

A la fin du projet (Question 4), nous aurons besoin de résoudre une équation non-linéaire. La première question du projet vise à implémenter deux méthodes de recherche de racine qui ont été vues au cours théoriques, à savoir la méthode de la bissection et la méthode de la sécante. Ces implantations doivent être le plus générique possible et donc pouvoir fonctionner pour tout type de fonction et avec le moins d'hypothèses de départ possibles. On demande d'implémenter dans un module **RechercheRacine** les deux fonctions

```
x, statut = secante(f, x0, x1, tol)
```

et

```
x, statut = bissection(f, x0, x1, tol)
```

qui permettent de rechercher la racine d'une fonction Python $f(x)$, avec une tolérance définie par la valeur **tol**, à partir de deux valeurs initiales **x0** et **x1** selon la méthode de la sécante et de la bissection. Il est primordial de respecter le format demandé ci-dessus étant donné que ces fonctions seront testées automatiquement. En sortie, les deux fonctions doivent retourner un objet de type "list" avec deux valeurs. Dans le cas où la recherche a correctement convergé et donne une approximation de la racine à la tolérance près, la première valeur **x** contient la valeur de la racine et la deuxième valeur **statut** contient la valeur 0. Dans le cas où l'utilisateur a rentré deux valeurs qui ne satisfont pas les hypothèses de la méthode, il vous est demandé d'afficher un message d'erreur et d'affecter à la variable de sortie **statut** la valeur 1. Dans ce cas, **x** contient n'importe quelle valeur. Dans le cas où la méthode de la recherche de racine ne converge pas (par exemple parce que la fonction n'a pas de racine), vous devez afficher un message d'erreur et affecter la valeur -1 à la variable **statut**. Dans ce cas, à nouveau, n'importe quelle valeur peut être affectée à la variable de sortie **x**.

Dans votre implémentation, prenez soin (i) de vérifier les hypothèses de chacune de ces méthodes et de prévoir un comportement adéquat du code dans le cas où ces hypothèses

ne sont pas rencontrées, de manière à fournir une réponse adaptée dans un maximum de cas, (ii) de veiller à réduire autant que possible le nombre d'appels de la fonction `f(x)` à l'intérieur des fonctions `secante` et `bisection`, et ce pour des raisons d'efficacité. L'utilisation du `profiler` de Spyder vous permettra d'analyser l'efficacité de votre code.

Question 2 : Pertes et gains de chaleur

La fonction $G(t)$ dans le modèle représente les flux de chaleur (pertes et gains) à travers l'enveloppe du bâtiment vers l'extérieur et les apports de chaleur provenant de l'activité au sein des bureaux (appareils électroniques, éclairages, activités humaines...). Ces pertes et gains sont donnés dans le tableau du fichier “PerteEtGain.txt” en W/m^2 pour une période de 24h représentant la moyenne des pertes et des gains horaires pour l'ensemble des journées sur un an. Dans ce fichier, la première ligne représente les heures de la journée et la deuxième ligne représente les flux de chaleur.

1. Importer les pertes et gains $G(t)$ du fichier texte dans Python et les temps t correspondants.
2. Déterminer une fonction interpolant ces données en justifiant le choix de la méthode.
3. Vérifier graphiquement que la fonction interpolée $G(t)$ obtenue reproduit les données de départ.

Question 3 : Mise en place de la modélisation

Établissez le modèle numérique permettant d'effectuer la simulation du système dans un module nommé `SimTABS`.

1. Définir dans Python une fonction `odefunction` qui reçoit en argument le temps t en heures et les valeurs courantes des températures en degrés Celsius dans un array de taille 5

$$T = [T_{room}, T_t, T_{cc}, T_{c1}, T_{c2}]^T \quad (6)$$

et fournit la dérivée des températures dT/dt en K/h selon les équations (1) à (5) pour la fonction $G(t)$ donnée dans le fichier “PerteEtGain.txt” et en supposant que l'on refroidit le bâtiment de 0 à 4h et que le système est ensuite mis à l'arrêt (scénario 1).

2. En utilisant cette fonction, implémenter la méthode d'Euler explicite pour résoudre le système d'équations différentielles sur l'intervalle de temps $[t_0, t_f]$ à partir des conditions initiales T_0 . On demande de programmer cette méthode dans un module `SimTABS` sous la forme d'une fonction

```
t, T = calculTemperaturesEuler([t0,tf], T0, h)
```

où $[t_0, t_f]$ est l'intervalle de résolution (en heures), T_0 est un array de taille 5 représentant les températures initiales en t_0 et h est le pas de temps (en heures). La sortie est un objet de type “list” composé des variables t et T . t est un array de n points où

la solution numérique est évaluée (n n'est pas imposé), T est un array de taille $5 \times n$ contenant les valeurs des températures en chaque point. Déterminer la solution sur l'intervalle $[0, 24h]$ en prenant comme conditions initiales $T_0 = [15, 15, 15, 15, 15]^\circ\text{C}$.

3. Utiliser la fonction `solve_ivp` de la librairie `scipy`. Comme à la question précédente, la résolution doit être programmée dans le module SimTABS sous la forme d'une fonction

```
t, T = calculTemperaturesIVP([t0,tf], T0, rtol)
```

où `rtol` est la tolérance relative de `solve_ivp`.

4. En considérant comme référence la solution obtenue avec `calculTemperaturesIVP` avec $\text{rtol} = 10^{-10}$, étudier la convergence de la solution obtenue avec `calculTemperaturesEuler` en fonction du pas de temps h . Sur base de cette étude, justifiez le choix du pas de temps pour la suite du projet. Pour les questions suivantes, seule la fonction `calculTemperaturesEuler` sera utilisée.
5. En supposant que la fonction décrivant les pertes et gains se répète chaque jour de même que les cycles de chauffage/refroidissement/mise à l'arrêt, calculer l'évolution des températures sur plusieurs jours jusqu'à l'état stationnaire défini par la condition

$$T_{room}(t_f) - T_{room}(t_f - 24) < 0.01^\circ\text{C}$$

où t_f indique la fin d'une journée. Tracer l'évolution des températures de la pièce et de la partie inférieure du béton en fonction du temps jusqu'à l'état stationnaire.

6. Comparez les évolutions de températures pour les scénarios de cycle de chauffage/refroidissement décrits à la Table 1 (le scénario 1 est le scénario de base).

TABLE 1 – Scénarios de chauffage (C)/refroidissement (R)/mise à l'arrêt (0).

Heure	Scénario 1	Heure	Scénario 2	Heure	Scénario 3
$[0, 4]$	R	$[0, 4]$	R	$[0, 12]$	C
$]4, 24]$	0	$]4, 13]$	C	$]12, 24]$	R
		$]13, 24]$	0		

Question 4 : Régulation automatisée du bâtiment

La température de confort du bâtiment est définie comme étant la moyenne entre la température de l'air dans la pièce T_{room} et celle de la surface du béton supposée égale à T_{c2} . Selon la norme EN15251, la température de confort doit être située entre 19.5°C et 24°C pendant les heures de bureau (8h-19h).

Tout d'abord, on considère les conditions initiales $T_0 = [15, 15, 15, 15, 15]^\circ\text{C}$, la fonction $G(t)$ donnée dans le fichier “PerteEtGain.txt”, un refroidissement du bâtiment de 0 à 4h,

TABLE 2 – Scénario de chauffage et de refroidissement pour la question 4.

Heure	Scénario 4
[0, 4]	R
]4, 4 + Δt]	C
]4 + Δt, 24]	0

une période de chauffe à partir de 4h du matin pendant une période Δt donnée et ensuite une mise à l'arrêt jusqu'à la fin du cycle de 24h (voir Table 2). La durée Δt est constante jour après jour. On demande ensuite de :

1. Définir dans Python une fonction permettant de déterminer le maximum de la température de confort T_{max} sur le premier cycle de 24h. Vérifiez graphiquement votre réponse.
2. Définir dans Python une fonction permettant de déterminer la période Δt nécessaire pour que T_{max} atteigne une valeur donnée T_{max}^d du maximum de température sur le premier cycle de 24h. Dans ce but, vous devez utiliser une des fonctions de la question 1, en justifiant le choix de la méthode et du/des paramètres numériques associés.
3. Définir dans Python une fonction permettant de déterminer la période Δt nécessaire pour que T_{max} atteigne une valeur donnée T_{max}^d du maximum de température sur le cycle de 24h lorsque le système est à l'état stationnaire. En choisissant $T_{max}^d = 24^\circ\text{C}$, vérifier si la norme EN15251 est respectée dans ces conditions.

TABLE 3 – Données numériques.

Variable	Valeur	Unités
C_{cc}	50	kJ/m ² K
C_{c1}	50	kJ/m ² K
C_{c2}	10	kJ/m ² K
C_{room}	12	kJ/m ² K
C_w	30	kJ/m ² K
R_{cc-c1}	0.05	m ² K/W
R_x	0.025	m ² K/W
R_{c2-cc}	0.02	m ² K/W
R_{r-s}	0.1	m ² K/W
R_{s-c2}	0.183	m ² K/W
R_w	0.15	m ² K/W

Consignes

- Le travail comporte un code de calcul Python et un rapport d'une longueur de 10 pages maximum (la page de garde n'est pas comptabilisée).
- Le code doit être correct et écrit par vous (ce que nous vérifierons à la présentation orale).
- Etant donné qu'un des objectifs du cours est la maîtrise des algorithmes numériques et de leur programmation en Python, il est interdit d'utiliser ChatGPT ou tout autre outil d'intelligence artificielle pour la création du code.
- Le code doit être soigné et commenté.
- Le code doit utiliser au maximum les possibilités vectorielles de `numpy`.
- Pour toute fonction, nous sommes susceptibles de vous demander de montrer un *profile* Python et de l'interpréter.

Critères d'évaluation

La note finale individuelle sera définie sur base de l'examen oral en prenant en compte la note de l'évaluation continue (milestones) ainsi que la note du rapport et du code Python. Un des objectifs principaux de ce cours étant l'apprentissage de la programmation des méthodes numériques en Python, il est indispensable que chaque étudiant et chaque étudiante réussisse l'examen oral, qui portera notamment sur la maîtrise de Python et de votre code, afin de réussir globalement.

Evaluation continue

Deux “milestones” permettront de vérifier votre état d'avancement en cours de projet.

- Mardi 18 février à 22h00 : vérification du tutoriel. Il s'agit d'une évaluation obligatoire mais formative, qui n'intervient pas dans la note finale.
- Dimanche 9 mars à 18h00 : vérification des questions 1 (fonctions `secante` et `bisection`) et 3 (fonctions `odefunction`, `calculTemperaturesEuler` et `calculTemperaturesIVP`). Le fichier “PerteEtGain.txt” doit être inclus dans la soumission. Le résultat de ce milestone interviendra dans la note finale.

Ces fonctions seront testées automatiquement avec Gradescope :

- L'accès à Gradescope se fait via eCampus, rubrique “outils du cours”.
- Il est primordial de respecter les consignes et le format demandé en termes de nom de fonction, de nom de module, de variables d'entrée et de sortie. Un non-respect des consignes sera sanctionné par une note de 0/10.
- Il est possible de charger un seul fichier Python ou une archive au format ZIP comprenant plusieurs fichiers.
- Seules les librairies `numpy`, `scipy` et `matplotlib` sont autorisées.
- Les soumissions seront ouvertes à partir du mercredi à 12h00 et vous aurez droit à 5 tentatives de soumission avant l'échéance. Au-delà de 5 tentatives, une pénalité

sera appliquée.

Rapport et code Python

Chaque groupe doit soumettre un rapport au format PDF et un fichier .zip comprenant les fichiers .py de votre programme sur Gradescope au plus tard pour le dimanche 6 avril à 18h00. Le nom du fichier .zip et le nom du fichier .pdf doivent respecter le format suivant : “NumeroGroupe_NomA_NomB_NomC.xxx” (exemple : l’archive “27_Dupond_Beckers_Bastin.zip” doit inclure le fichier “27_Dupond_Beckers_Bastin.pdf”).

- La longueur du rapport ne peut dépasser 10 pages (la page de garde n'est pas comptabilisée) et ne doit pas comporter d'introduction.
- Pour chaque question, les résultats obtenus doivent être illustrés.
- La justification des choix numériques est très importante. Pensez à expliquer les choix qui vous ont semblé cruciaux.
- La forme du rapport est prise en compte. Il est recommandé de suivre les règles de bonne pratique pour la réalisation d'un rapport scientifique qui sont présentées dans le podcast correspondant. Le nombre de pages étant limité, il est inutile de répéter l'énoncé. Allez donc à l'essentiel.
- La qualité du code (efficacité et soin) est également considérée dans l'évaluation.

Examen oral

L'examen oral est **individuel** et dure 10 minutes. Vous devez faire une démonstration du programme de votre groupe et répondre à des questions supplémentaires. Les éléments suivants seront pris en considération :

- la maîtrise de Python en tirant au sort dans une liste de questions disponibles avant l'examen ;
- la maîtrise du programme réalisé par votre groupe en tirant au sort dans une liste de questions disponibles avant l'examen (vous devez être capable d'exécuter le code, de le modifier et d'expliquer son fonctionnement) ;
- les justifications et éclaircissements par rapport aux choix réalisés et aux résultats présentés dans le rapport et dans le code ;
- la maîtrise des notions théoriques vues au cours en lien avec votre travail et la maîtrise générale du projet.

Deuxième session

Les groupes qui, en première session, ont obtenu pour le rapport et le code une note suffisante sont, s'ils le souhaitent, dispensés de remettre un nouveau code et un nouveau rapport lors de la session de septembre. Dans ce cas, seul l'oral doit être représenté et compte pour 100% de la note finale. Pour les autres groupes, un nouveau code et un nouveau rapport doivent être remis 5 jours avant l'examen oral.

Références

- [1] Sourbon M., De Herdt R., Van Reet, T., Van Passel W., Baelmans M., and Helsen L., 2009, “Efficiently produced heat and cold is squandered by inappropriate control strategies : A case study”, Energy and Buildings, 41, 1091-1098.
- [2] EN 15251, 2007, Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics, CEN.