

Promotion 2019
5^e année, Majeure IBO

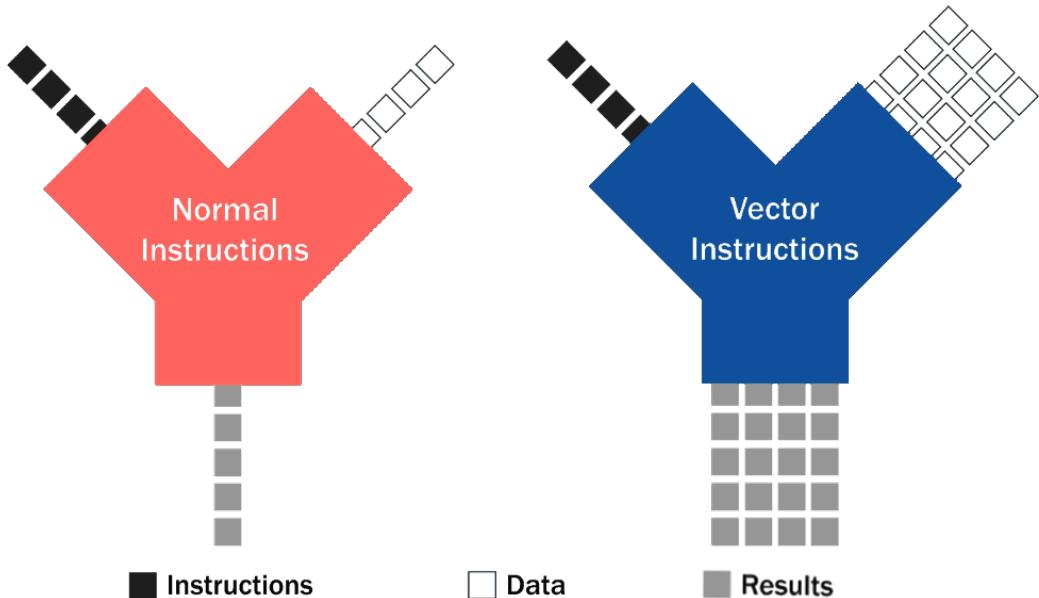
CSTB
DA2E
Marne-la-Vallée, France

Tuteur école :
Mme Sonia DJEBALI
Enseignante en informatique, ESILV

Tuteur entreprise :
M. Jérémie EL BEZE
Chef de projet R&D, CSTB

Développement Python et l'Optimisation

ZHANG Hao



Remerciements

Avant tout développement sur cette expérience professionnelle, je tiens à remercier tous les membres de la Direction Analyse et Études économiques (DA2E) qui m'a ont accueilli et fait découvrir toute l'entreprise.

Je tiens à remercier tout particulièrement mon tuteur d'entreprise de stage Monsieur Jérémy EL BEZE, pas seulement pour m'avoir accepté par ce stage, aussi pour avoir pris pas mal de temps pour m'expliquer les notions principales de son programme et discuter avec moi de l'optimisation des formules économiques. Pendant mon stage, il m'a laissé pas mal de temps libre à réfléchir l'optimisation de ce programme, et de points bloqués à discuter ce programme quand j'en ai besoin. Grâce à sa compression, j'ai pu apprendre dans d'excellentes conditions avec un accompagnement de qualité.

Je remercie Mme Lydia CHIBOUT et M. Thamer MECHARNIA. Quand j'ai rencontré des difficultés et des goulots d'étranglement, ils m'ont fourni beaucoup d'opinions professionnelles et m'ont aidé à résoudre de nombreux problèmes.

Je voudrais tout particulièrement remercier M. Antoine BREITWILLER. Lorsque je me suis senti désespéré en raison de l'efficacité du programme en matière de calcul, ses conseils en matière de programmation vectorisée m'ont inspiré. Merci d'avoir accepté de prendre le temps de me guider en tant que stagiaire, même s'il ne fait pas partie du même département que moi.

Aussi je tiens à remercier mon tuteur d'école, Mme Sonia DJEBALI, ainsi que toutes les personnes m'ayant permis de profiter d'un stage aussi intéressant et complet.

Résumé

Du 6 avril 2019 au 5 novembre 2019, j'ai eu la chance d'effectuer un stage au sein de la Direction Analyse et Études économiques (DA2E) de l'entreprise CSTB (Centre Scientifique et Technique du Bâtiment). Au cours de ce stage et à travers ma mission du développement Python et l'optimisation, j'ai pu découvrir pleinement des pratiques de programmation en Python, des techniques pour la vectorisation et l'optimisation ainsi que des connaissances corrélatives sur la substitution énergétique, l'efficacité énergétique, la modélisation prospective et la minimisation des coûts.

Et plus largement, ce stage a été l'opportunité pour moi d'appréhender l'application du langage Python sur la programmation orientée objet (POO), la modélisation de programme, ainsi que cela m'a également permis de mieux comprendre la programmation inclut l'algorithme, la simplification, l'optimisation et l'accélération. Afin de résoudre la vitesse de calcul et l'efficacité du programme, j'ai appris et mis en œuvre le concept de vectorisation, combinant ainsi les avantages de la programmation orientée objet. Non seulement j'ai réussi à réduire le temps de calcul du programme, mais j'ai également maîtrisé ces techniques plus efficacement et profondément.

Au-delà d'enrichir mes connaissances dans le domaine la programmation en Python, ce stage m'a permis de comprendre dans quelle mesure mon parcours professionnel continue de se préciser, en effet à la suite de ce stage je souhaite continuer mes études dans le domaine de data science et de machine learning en bien maîtrisant la manière de modélisation et de vectorisation ainsi que d'un bon niveau de data analyse. Simultanément, je souhaite aussi découvrir plus de techniques et de notions pour machine learning ou deep learning, que ce soit dans l'étude de l'algorithme ou bien dans la réalisation de l'intelligence artificielle de demain.

Mon stage chez CSTB a consisté essentiellement en traduction d'un programme VBA en Python, et le point le plus important, c'est d'optimiser ce programme python surtout pour réduire le temps de calcul de ce programme. Au cours de mon stage, bien que j'ai également bénéficié de l'aide de nombreux collègues expérimentés en programmation python ou en analyse de données, la plupart du temps, j'ai terminé l'ensemble du programme et optimisé en étudiant et en pensant seul. Bien que tout le processus ait été un peu ennuyeux, long et même un peu tergiversant, j'en ai aussi beaucoup appris et j'ai été en mesure de mieux saisir cette connaissance, qui est également devenue l'une des parties les plus significatives de tout mon processus de stage.

Mots-clés : Python, modélisation, simulation prospective , optimisation, vectorisation

Summary

From April 6, 2019, to November 5, 2019, I had the chance to do an internship in the Department Analysis and Economic Studies (DA2E) of the company CSTB (Scientific and Technical Center Building). During this internship and through my mission of Python development and optimization, I was able to fully discover programming practices in Python, techniques for vectorization and optimization as well as correlative knowledge on energy substitution, energy efficiency, prospective modeling, and cost minimization.

And more broadly, this internship was the opportunity for me to apprehend the application of the Python language on object-oriented programming (OOP), the program modeling, as well as it also allowed me to better understand the programming includes algorithm, simplification, optimization, and acceleration. In order to solve the computational speed and efficiency of the program, I learned and implemented the concept of vectorization, thus combining the advantages of object-oriented programming. Not only did I reduce the computing time of the program, but I also mastered these techniques more effectively and deeply.

Beyond enriching my knowledge in the field of programming in Python, this internship allowed me to understand how my career continues to be clarified, indeed as a result of this internship I wish to continue my studies in the domain of data science and machine learning in well mastering the way of modeling and vectorization as well as a good level of data analysis. Simultaneously, I also want to discover more techniques and concepts for machine learning or deep learning, whether in the study of the algorithm or in the realization of the artificial intelligence of tomorrow.

My internship at CSTB consisted mainly of translating a VBA program into Python, and the most important point is to optimize this python program especially to reduce the calculation time of this program. During my internship, although I also benefited from the help of many colleagues experienced in python programming or data analysis, most of the time I completed the entire program and optimized by studying and thinking alone. Although the whole process was a bit boring, time-consuming and even a bit dull, I also learned a lot and was able to better understand this knowledge, which has also become one of the most significant parts of all my internship processes.

Keywords : Python, modeling, prospective simulation, optimization, vectorization

Table des matières

1	Introduction	5
2	Monographie de l'entreprise	7
2.1	Missions et activités	7
2.2	Positionnement et stratégie	7
2.3	Direction de l'Analyse et des Études économiques	8
2.4	Contexte et objectifs du stage	9
2.5	Contexte de la recherche connexe	9
3	Mes missions	12
3.1	Découverte du modèle Zephyr-chaleur	12
3.1.1	Mécanismes de modélisation	12
3.1.2	Présentation du programme original	12
3.2	Données utilisées dans le projet	16
3.2.1	Extraction et préparation des données d'origine	16
3.2.2	Présentation des tableaux	17
3.3	Implémentation de la programmation Python	19
3.3.1	Idées de développement	19
3.3.2	Quelques mots sur la POO	20
3.3.3	Opération pratique	20
3.3.4	Résultats obtenus	21
3.3.5	Défauts de conception	23
3.4	Implémentation de la programmation Python avec la vectorisation	23
3.4.1	Quelques mots sur la vectorisation	23
3.4.2	Retraitement des formules	24
3.4.3	Intégration des données d'original	25
3.4.4	Structure du programme	27
3.4.5	Programmation vectorisée comme un puzzle	29
3.4.6	Effets opérationnels et recommandations futures	30
4	Conclusion	32

1 Introduction

Du 6 avril 2019 au 5 novembre 2019, j'ai eu la chance d'effectuer un stage au sein de la Direction Analyse et Études économiques (DA2E) de l'entreprise CSTB (Centre Scientifique et Technique du Bâtiment). Au cours de ce stage et à travers ma mission du développement Python et l'optimisation, j'ai pu découvrir pleinement des pratiques de programmation en Python, des techniques pour la vectorisation et l'optimisation ainsi que des connaissances corrélatives sur la substitution énergétique, l'efficacité énergétique, la modélisation prospective et la minimisation des coûts.

Et plus largement, ce stage a été l'opportunité pour moi d'appréhender l'application du langage Python sur la programmation orientée objet (POO), la modélisation de programme, ainsi que cela m'a également permis de mieux comprendre la programmation inclut l'algorithme, la simplification, l'optimisation et l'accélération. Afin de résoudre la vitesse de calcul et l'efficacité du programme, j'ai appris et mis en œuvre le concept de vectorisation, combinant ainsi les avantages de la programmation orientée objet. Non seulement j'ai réussi à réduire le temps de calcul du programme, mais j'ai également maîtrisé ces techniques plus efficacement et profondément.

Au-delà d'enrichir mes connaissances dans le domaine la programmation en Python, ce stage m'a permis de comprendre dans quelle mesure mon parcours professionnel continue de se préciser, en effet à la suite de ce stage je souhaite continuer mes études dans le domaine de data science et de machine learning en bien maîtrisant la manière de modélisation et de vectorisation ainsi que d'un bon niveau de data analyse. Simultanément, je souhaite aussi découvrir plus de techniques et de notions pour machine learning ou deep learning, que ce soit dans l'étude de l'algorithme ou bien dans la réalisation de l'intelligence artificielle de demain.

Mon stage chez CSTB a consisté essentiellement en traduction d'un programme VBA en Python, et le point le plus important, c'est d'optimiser ce programme python surtout pour réduire le temps de calcul de ce programme. Au cours de mon stage, bien que j'ai également bénéficié de l'aide de nombreux collègues expérimentés en programmation python ou en analyse de données, la plupart du temps, j'ai terminé l'ensemble du programme et optimisé en étudiant et en pensant seul. Bien que tout le processus ait été un peu ennuyeux, long et même un peu tergiversant, j'en ai aussi beaucoup appris et j'ai été en mesure de mieux saisir cette connaissance, qui est également devenue l'une des parties les plus significatives de tout mon processus de stage.

En général, mon processeur de stage peut être divisé en trois périodes :

- En première période, j'ai traduit le programme VBA à Python.
- En deuxième période, j'ai commencé à réfléchir et chercher des solutions pour optimiser ce programme.
- En troisième période, j'ai pratiqué et achevé de la réalisation de la vectorisation du programme.

Mon tuteur d'entreprise de stage Monsieur Jérémy EL BEZE est le chef de projet de recherche et développement de la Direction Analyse et Études économiques de CSTB, il s'occupe principalement de la conception et du développement de logiciels utilisant les dernières technologies en matière d'apprentissage automatique et de machine learning.

galement le travail d'Analyse et modélisation économiques. Le but de mon stage est de transduire son programme VBA en Python. Ce programme est un modèle de simulation prospectif nommé Zephyr-Chaleur qui est dérivé de sa thèse. Il s'agit d'un outil de simulation prospective basé sur une approche technico-économique et centré sur la minimisation de GHC (Global Heating Costs) d'un grand nombre de logements types, représentatifs du parc résidentiel français. Pendant mon stage, il a pris pas mal de temps pour m'expliquer les notions principales de son programme et discuter avec moi de l'optimisation des formules économiques. Également, il m'a laissé pas mal de temps libre à réfléchir l'optimisation de ce programme, et de points bloqués à discuter ce programme quand j'en ai besoin. Grâce à sa compression, j'ai pu apprendre dans d'excellentes conditions avec un accompagnement de qualité.

Ce rapport est arrangé de la manière suivante. Dans un premier je vous ferai une présentation générale de l'entreprise, la direction, l'objectif de stage et le contexte des recherches relatives. Par la suite je vous décrirai le modèle Zephyr-Chaleur et les données d'original. Enfin, je développerai ce que j'ai fait pour un programme pur Python et pour un programme de vectorisation en Python.

2 Monographie de l'entreprise

2.1 Missions et activités

Le CSTB, Centre Scientifique et Technique du Bâtiment a été créé en 1947 pour accompagner la reconstruction d'après-guerre en France. Il fut notamment un outil charnière du ministère de la Reconstruction et de l'Urbanisme (MRU) après la guerre de 1939-1945.

Le CSTB a pour mission de garantir la qualité et la sécurité des bâtiments. Il rassemble pour cela des compétences pluridisciplinaires pour développer et partager les connaissances scientifiques et techniques déterminantes. Il accompagne les acteurs dans le cycle de l'innovation de l'idée au marché et dans la transformation du monde du bâtiment en lien avec les transitions environnementale, énergétique et numérique. Le CSTB exerce 5 activités clés : la recherche et expertise, l'évaluation, la certification, les essais et la diffusion des connaissances. Son champ de compétences couvre les produits de construction, les bâtiments et leur intégration dans le quartier et la ville.

2.2 Positionnement et stratégie

Atteindre les objectifs ambitieux de la transition énergétique et numérique, en soutenant l'innovation durable, impose de transformer le secteur du bâtiment. Disposant de compétences pointues, le Centre Scientifique et Technique du Bâtiment poursuit l'adaptation de ses services, de ses moyens et de son organisation pour accompagner cette transformation auprès de l'ensemble des acteurs. En lien avec l'évolution sociétale et l'attente de « mieux vivre » des citoyens, il porte les enjeux techniques de performance énergétique, environnementale, mais aussi de santé, d'accessibilité et de confort du bâtiment. Fort de son expertise BIM (Building Information Modeling), il est engagé dans l'enrichissement et le déploiement de la maquette numérique auprès des professionnels de la construction et de l'aménagement urbain.

3 axes stratégiques

La performance globale des bâtiments La mise en œuvre de la transition écologique et énergétique, la préservation de la santé, la prévention des risques, la maîtrise des coûts de construction et de rénovation, l'usage et l'adaptabilité des bâtiments aux évolutions sociales nécessitent de développer la performance globale des bâtiments.

Un accompagnement renforcé des acteurs dans le processus d'innovation Le meilleur accès des entreprises à de nouveaux marchés et le développement de l'emploi nécessitent un accompagnement renforcé des acteurs engagés dans le processus d'innovation.

Une approche intégrée à l'échelle des villes et des territoires L'émergence et le développement de villes durables s'appuyant sur les performances des bâtiments au service de la qualité de vie des citoyens nécessitent une approche intégrée jusqu'à l'échelle des villes et des territoires.

Pour mettre en œuvre ces 3 axes stratégiques, le CSTB organisera ses activités et moyens autour de :

- **5 priorités scientifiques et techniques** : Énergie-Environnement, Santé-Confort, Maîtrise des risques, Numérique, Usages, Économie, Mutations des filières, permettant de porter une analyse scientifique et technique multicritères couvrant l'ensemble des enjeux de développement durable, mais aussi interdisciplinaire, qui font la spécificité d'un centre scientifique et technique tel que le CSTB ;
- **2 niveaux d'intégration** : d'une part des composants intégrés à l'ouvrage « bâtiment », d'autre part des bâtiments intégrés dans la ville. On veillera ainsi à intégrer à ces deux échelles la dimension temporelle par la prise en compte de l'ensemble du cycle de vie ;
- **5 activités** : Recherche et expertise, Évaluation, Certification, Essais, Diffusion des connaissances.

5 priorités scientifiques et techniques

Énergie-Environnement Améliorer et garantir la performance énergétique et environnementale de l'espace bâti repose sur une approche globale en cycle de vie, prenant en compte les enjeux d'économie circulaire.

Santé-Confort Étudier les ambiances, tant urbaines qu'à l'intérieur des bâtiments, identifier les facteurs de risques potentiels pour les populations et proposer des solutions de protection et d'amélioration.

Maîtrise des risques Développer une approche multirisques de l'acte de construire et de rénover, pour appréhender dans une seule approche de conception intégrant tous les types de risques.

Numérique Développer des outils numériques collaboratifs multicritères pour les professionnels de l'aménagement, conception, construction et rénovation, pour accompagner l'évolution des pratiques.

Usages, Économie, Mutations des filières Repenser bâtiments et lieux de vie en réponse aux nouveaux usages, attentes et contraintes environnementale, nécessite d'accompagner l'évolution des processus d'aménagement, conception et gestion de la construction.

2.3 Direction de l'Analyse et des Études économiques

La Direction de l'Analyse et des Études économiques (DA2E) est fortement investie dans la Recherche et l'Expertise dans le domaine de la transition énergétique et environnementale dans les bâtiments.

Dans le cadre d'un projet de recherche dédié, la DA2E :

- développe différents outils de modélisation permettant l'optimisation des actions de rénovations énergétiques (de l'échelle locale à l'échelle nationale) et l'évaluation de l'impact des instruments de politiques publiques sur les investissements en efficacité énergétique et en réduction d'émissions de CO₂ ;
- cherche à quantifier l'effet des différentes barrières à l'investissement sur la réalisation de la transition énergétique et à modéliser les comportements d'usage de l'énergie dans les secteurs résidentiel et tertiaire ;
- développe des méthodologies d'identification et d'analyse des déterminants de la valorisation des externalités liées aux rénovations énergétiques (valeur patrimoniale, santé, confort, etc.) ;
- accompagne des collectivités et des gestionnaires de parcs immobiliers dans leurs stratégies de construction/rénovation.

2.4 Contexte et objectifs du stage

La Direction de l'Analyse et des Études économiques du CSTB a développé ces dernières années un modèle d'optimisation des investissements en rénovation énergétique. Ce modèle repose sur une modélisation technico-économique du parc résidentiel français et permet de déterminer les combinaisons d'opérations de rénovation (isolation des différentes composantes de l'enveloppe thermiques des bâtiments, remplacement des systèmes de chauffage) permettant de minimiser le coût d'atteinte de différentes cibles de réduction d'émissions à l'horizon 2035 sous diverses contraintes (plafond annuel de rénovation, enveloppe annuelle d'investissement). Cet outil est mobilisé à la fois dans le cadre d'études commandées par la Direction de l'Habitat, de l'Urbanisme et des Paysages et dans des missions d'appui à l'élaboration de plan de gestion de Patrimoine.

Afin d'améliorer l'efficacité opérationnelle de ce modèle, d'optimiser la conception de son programme et de mieux potentiellement coopérer avec d'autres modèles d'analyse d'autre direction de CSTB, DA2E espère réécrire le programme VBA original dans un programme Python et effectuer une optimisation plus appropriée.

2.5 Contexte de la recherche connexe

L'amélioration de l'efficacité énergétique est un défi majeur pour atteindre les objectifs de transition énergétique de l'UE (en particulier de la France). Les politiques énergétiques actuelles doivent atteindre trois objectifs : décarboniser la production d'énergie, garantir la sécurité d'approvisionnement et maintenir la compétitivité des prix de l'énergie. Les énergies renouvelables et l'efficacité énergétique sont les principaux outils pour atteindre ces objectifs, mais si les énergies renouvelables peuvent fournir une énergie sobre en carbone, ces technologies ne sont pas suffisamment évolutives, rentables et prévisibles. Par conséquent, nous devons améliorer l'efficacité de notre système énergétique^[1] : en réduisant la demande énergétique finale, nous pouvons mieux contrôler les prix de l'énergie, accroître la compétitivité de l'industrie et réduire la dépendance vis-à-vis des producteurs d'énergie externes. Peut réduire les émissions de gaz à effet de serre^[2].

Dans l'ensemble de l'UE, 26% de la consommation d'énergie finale est utilisée pour les bâtiments résidentiels^[3]. Pour la France, cette part est plus élevée, à 28%. Après les transports, le secteur de la construction résidentielle est le plus gros consommateur d'énergie finale. Une consommation d'une telle importance a attiré l'attention des gens : le premier est la sécurité d'approvisionnement, car l'UE importe plus de 50% de l'énergie. Deuxièmement, le réchauffement climatique : la Commission européenne a fixé des objectifs ambitieux visant à réduire les émissions de gaz à effet de serre en Europe de 40% d'ici 2030 par rapport aux niveaux de 1990 et de 80 à 95% d'ici 2050. Ces objectifs de CO₂ sont liés aux objectifs d'efficacité énergétique. Pour atteindre ces objectifs ambitieux, la Commission européenne et le gouvernement français ont mis en place diverses politiques publiques et ont débattu de l'efficacité de ces politiques^[4].

Le secteur de la construction représente un très grand domaine d'efficacité énergétique, en particulier les bâtiments résidentiels. En Europe, environ 70% de l'énergie des ménages est utilisée pour le chauffage de locaux et cette demande peut être considérablement réduite. Dans la chaîne de valeur énergétique, trois actions peuvent être envisagées pour accroître l'efficacité des maisons et des bâtiments. Premièrement, il est possible de réduire la consommation d'énergie primaire et les émissions de gaz à effet de serre en convertissant l'énergie électrique en biomasse. Deuxièmement, pour une même demande d'énergie, la consommation finale d'énergie peut être réduite en remplaçant la chaudière par une chaudière plus efficace. Troisièmement, les besoins en énergie et en chauffage peuvent être réduits en rénovant l'isolation d'un bâtiment. Si les deux premières actions réduisent la consommation d'énergie en améliorant la conversion d'énergie, seule la réparation thermique des enveloppes de bâtiment peut réellement réduire la demande en énergie pour les utilisations finales^[5]. En fait, les vieilles maisons et les bâtiments ont subi d'importantes pertes de chaleur dans l'enveloppe et nécessitent donc un chauffage plus puissant. Il existe plusieurs technologies pour isoler les composants encapsulants (murs, fenêtres, toits et planchers), ce qui peut considérablement réduire les besoins en énergie. L'utilisation de solutions technologiques de pointe peut à terme transformer les bâtiments en bâtiments à énergie quasi nulle^[6]. Les travaux en cours sont axés sur la remise à neuf des enveloppes afin d'améliorer l'efficacité d'utilisation finale.

Afin de répondre à cette question, un modèle de simulation prospective technico-économique est développé dans la thèse de mon tuteur d'entreprise, Monsieur Jérémy EL BEZE. Procédant d'une approche Bottom-up, le modèle Zephyr-chaleur se caractérise par une représentation très fine du parc de logements français^[7]. Ce modèle introduit les contraintes temporelles résultant du cycle de renouvellement des systèmes de chauffage et d'un plafond annuel de logements pouvant faire l'objet de travaux d'isolation. La prise en compte de ces contraintes permet de simuler l'évolution prospective du parc de résidentiel français en fonction de différents objectifs de réduction d'émissions sur la période 2015-2035 et en supposant que les voies coûts-efficaces sont systématiquement suivies. Les résultats de simulation permettent d'évaluer la contribution des facteurs de substitution et d'efficacité énergétiques (Figure 1) en fonction des objectifs de décarbonation du parc résidentiel.

Les simulations de cette thèse soulignent un rôle d'autant plus important de la substitution dans l'atteinte des objectifs de réduction d'émissions que ceux-ci sont ambitieux. À l'inverse, loin de

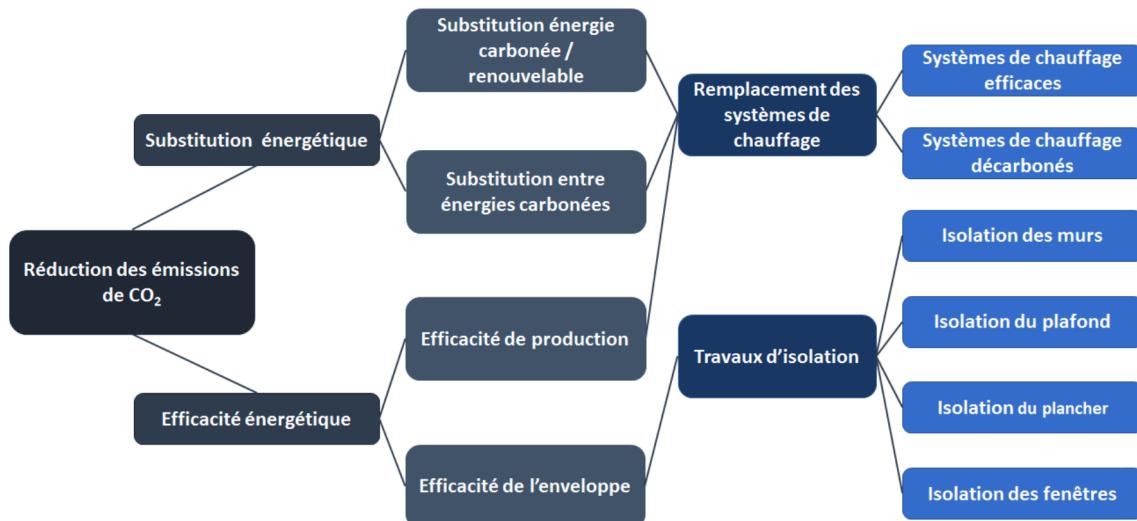


Figure 1 – Efficacité, substitution et opérations de rénovation énergétique

ce que suggéreraient les objectifs de performance énergétique adoptés par les pouvoirs publics, le rôle de l'amélioration de l'efficacité de l'enveloppe des bâtiments semble jouer un rôle qui devient secondaire dans la plupart des cas quand on augmente l'ambition climatique. Dans de nombreux cas, le niveau moyen coût efficace d'isolation des logements tend même à décroître lorsque les objectifs de décarbonation du parc s'élèvent. Ces résultats conduisent à discuter les priorités des politiques conduites en France à l'égard du secteur résidentiel et l'accent mis sur la rénovation du bâtiment ancien. Pour viser la réduction massive des émissions de CO₂, un rééquilibrage des politiques publiques au profit de la substitution permettrait sans doute de réduire les coûts et donc de renforcer la probabilité de réaliser des objectifs qui n'ont été jusqu'à présent ni atteints ni même approchés.

3 Mes missions

3.1 Découverte du modèle Zephyr-chaleur

3.1.1 Mécanismes de modélisation

Le modèle Zephyr-Chaleur prolonge et approfondit l'approche précédente. Il s'agit d'un outil de simulation prospective basé sur une approche technico-économique et centré sur la minimisation de GHC (Global Heating Costs) d'un grand nombre de logements types, représentatifs du parc résidentiel français. Il procède d'une démarche bottom-up : l'évolution des caractéristiques physiques, techniques et économiques du stock de bâtiments résulte de l'agrégation des opérations menées au niveau des logements types. Suivant la logique précédemment empruntée, ce modèle n'est pas développé dans une perspective de prévision de l'évolution du parc résidentiel, mais vise à simuler d'hypothétiques transformations du parc de logements existants dans un cadre où celles-ci seraient uniquement guidées par la minimisation du GHC de manière à explorer les l'évolution des facteurs d'efficacité et de substitution énergétiques permettant d'opérer à moindre coût (Figure 2).

Le développement d'un modèle de simulation permet d'étudier la dynamique de transformation du parc résidentiel. La minimisation du GHC n'est donc plus réalisée sur une période unique. Elle est effectuée dans un cadre temporel donné, sur la base d'un pas de temps annuel. Chaque logement type débute une année n dans une configuration donnée en termes de composition de l'enveloppe thermique et d'équipement de système de chauffage et opère dans cette configuration tout au long de l'année. Puis, en fin d'année n , le GHC est minimisé sur un nouvel horizon temporel ($n + 1$ à $n + 40$) menant, ou non, à un remplacement du système de chauffage et/ou à une amélioration de certains composants de l'enveloppe thermique. Ces potentielles mesures sont supposées être mises en œuvre immédiatement tant et si bien que chaque logement type entame l'année $n + 1$ dans sa nouvelle configuration. Ce cycle est répété année après année sur une période de vingt ans, simulant ainsi l'évolution du parc de logement de 2015 à 2035.

3.1.2 Présentation du programme original

Zephyr-chaleur est un fichier xlsm d'environ 180 Mo, contenant de nombreuses macros VBA complexes. Bien que la taille ne soit que d'environ 180 Mo, étant donné que le fichier xlsm est essentiellement un package de fichier compressé, une fois le fichier ouvert normalement, il faudra environ 1,4 Go de mémoire sur l'ordinateur et la préparation du fichier prendra beaucoup de temps.

Le fichier xlsm se compose de trois grandes tableaux, les tableaux « 2015 », « HSD » et « Traitement 1 ». Parmi celles-ci, le tableau « 2015 » présente l'ensemble des données d'origine et des résultats de calcul pour un total de 26 logements dans une année, une zone et une catégorie de logement spécifiées (y compris le type de bâtiment spécifié, la taille et l'année d'utilisation du système de chauffage) pour les trois principaux types de logements. Chacun de ces types résidentiels a un type différent de système de chauffage (Figure 3).

Parmi eux, la première section du tableau présente les informations de calcul détaillées de ces

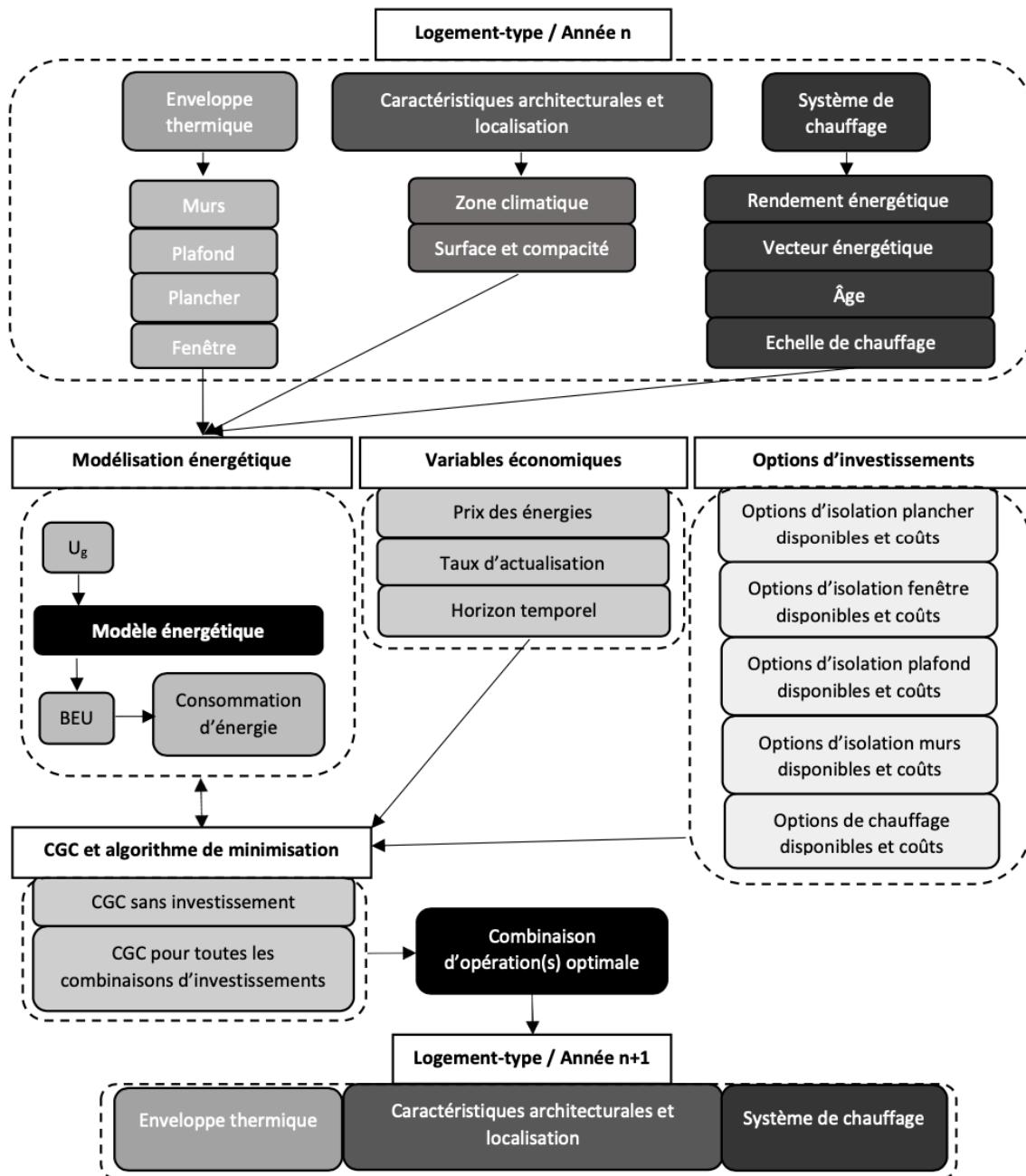


Figure 2 – Représentation du fonctionnement du modèle au niveau d'un logement type sur une année

Figure 3 – Le tableau « 2015 »

26 maisons dans le cadre d'un système d'isolation spécifié, y compris les principaux résultats de calcul d'Ug (the overall heat transfer coefficient), UEN (Useful Energy Needs) et GHC (Global Heating Costs). Cette section présente également les résultats de la valeur minimale de la GHC. si le système de chauffage et/ou d'isolation existant est optimisé.

Les deuxième, troisième et quatrième sections du tableau montrent les résultats des valeurs spécifiques de la GHC des trois grands types de logements (CID (Collective Dwellings with Individual heating system), CCD (Collective Dwellings with Collective heating system) et ID (Individual Dwellings) les résultats prévus pour chaque colonne représentant les différents types de systèmes de chauffage des mêmes logements dans la première section pour chaque type d'isolation.

La cinquième section du tableau fournit des informations spécifiques sur l'isolation des différents systèmes d'isolation des murs, des fenêtres, du toit et du sol de la maison, y compris le type, le transfert de chaleur et le coût.

En fait, le tableau « 2015 » comprend essentiellement toutes les données d'origine et les résultats de calcul liés aux paramètres résidentiels. Ce qui précède n'introduit que les cinq parties importantes présentées dans l'image, et non l'ensemble du tableau, et les données présentées sont loin d'indiquer le montant du calcul de ce tableau, c'est uniquement à titre de référence pour comprendre la structure générale du programme.

Le tableau « HSD » enregistre les facteurs de prix de tous les systèmes de chauffage qui changent en fonction de la date ou de l'année du système de chauffage pendant le processus de calcul complet, tels que le coût de la maintenance du système et le coût du remplacement

des systèmes sans gaz, etc. (Figure 4).

Figure 4 – Le tableau « HSD »

Le tableau « Traitement 1 » est la présentation finale de l'ensemble du modèle, divisé en 20 groupes sur 20 ans. Les résultats prévus de 32 760 logements appartenant à trois catégories résidentielles sont regroupés dans des colonnes différentes : chaque colonne enregistre le GHC du système de chauffage d'origine pour le système résidentiel, le GHC après modification du système d'isolation en partenariat avec le système de chauffage et le solde du GHC après le remplacement du système. Et ainsi de suite. En plus de cela, il fournit également des informations sur des systèmes de chauffage spécifiques et/ou des systèmes d'isolation correspondant aux modifications. Lors du changement d'année, le programme sera trié par ordre décroissant en fonction de la valeur de solde du GHC minimum. Les bâtiments résidentiels répondant aux exigences changeront le système de chauffage et/ou d'isolation correspondant. Les logements restants conserveront le système de chauffage et/ou le système d'isolation d'origine pour les calculs de l'année suivante (Figure 5).

Car le fichier Zephyr-chaleur contient toutes les données d'origine, les formules de calcul, les résultats de calcul de chaque processus de calcul et de nombreuses macros complexes. De mon tuteur d'entreprise, l'auteur du programme, étant un économiste plutôt qu'un programmeur professionnel, le programme dans son ensemble est trop compliqué pour les non-auteurs eux-mêmes, principalement à cause des points suivants :

- Les données d'origine ne sont pas distinguées des résultats ou des résultats excessifs produits par le processus de calcul, entraînant une réduction de l'évolutivité du logiciel ;
- Trop de macros utilisées pour le copier-coller et un grand nombre de processus de calcul rendent difficile la localisation précise de la position des données et de l'ordre des

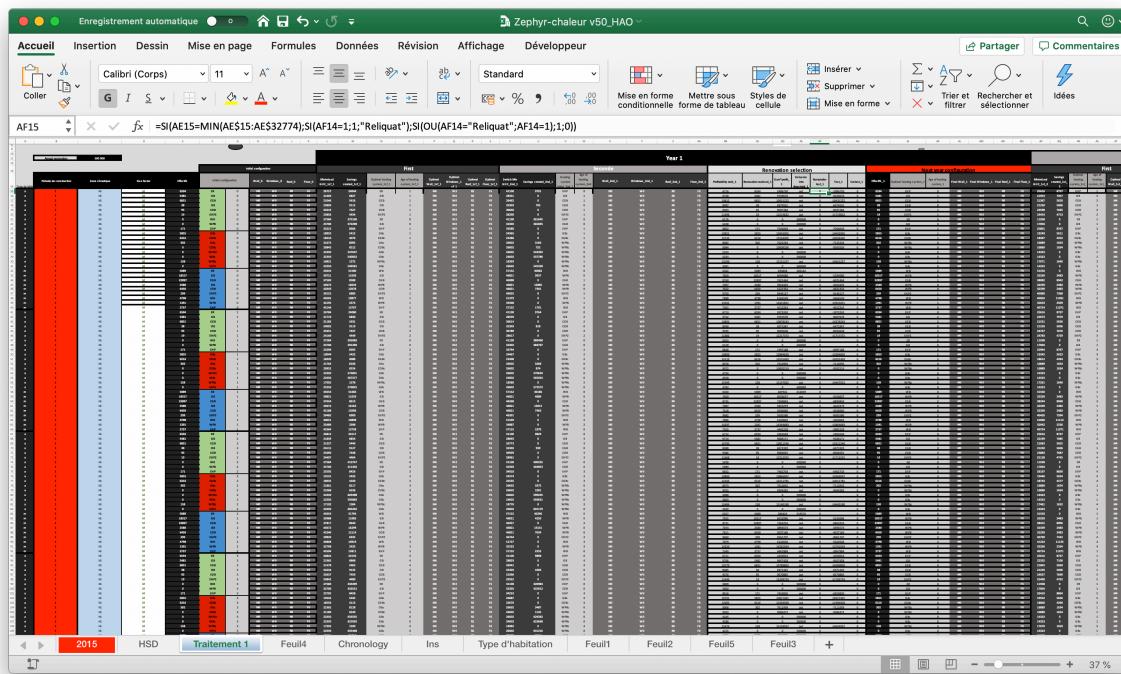


Figure 5 – Le tableau « Traitement 1 »

opérations ;

- De plus, il n'est pas facile de lire et de comprendre le programme par les macros ou les formules de chaque cellule ;
- La structure légèrement chaotique réduit considérablement la lisibilité du programme.

Par conséquent, dans toute la phase de programmation, tous les codes ne sont pas directement traduits conformément au code VBA de ce mode, mais convertis à partir de l'explication de mon tuteur et de la formule de calcul de cellules individuelles. Bien que le processus soit long et compliqué, il peut m'aider à mieux comprendre et à maîtriser l'ensemble du programme, afin de pouvoir le modifier et l'optimiser ultérieurement.

3.2 Données utilisées dans le projet

3.2.1 Extraction et préparation des données d'origine

Comme j'ai mentionné ci-dessus, étant donné que les données d'origine utilisées par le programme d'origine et les résultats du calcul se chevauchent trop dans la distribution, si elles sont utilisées directement, cela causera un certain degré d'obstacles au positionnement des données du nouveau programme. Et parce que le programme d'origine contient de nombreuses macros compliquées, son ouverture prend trop de temps et l'ordinateur utilise trop de mémoire. Par conséquent, avec l'aide de mon tuteur, nous avons réextrait toutes les données d'origine et les avons transférées dans un nouveau fichier xlsx nommé « Input_Data ».

Le fichier « Input_Data » comprend dix tableaux : « Wall Investment » (Table 1), « Window Investment » (Table 2), « Roof Investment » (Table 3), « Floor Investment » (Table 4), « Eco-

Tableau 1 – Wall Investment	Tableau 2 – Window Investment	Tableau 3 – Roof Investment
Collective dwelling	Collective dwelling	Collective dwelling
Wall Investment	Window Investment	Roof Investment
Um	Uw	Ur
Price including Tax	Price including Tax	Price including Tax
Individual dwelling	Individual dwelling	Individual dwelling
Wall Investment	Window Investment	Roof Investment
Um	Uw	Ur
Price including Tax	Price including Tax	Price including Tax
Tableau 4 – Floor Investment	Tableau 5 – Economic Hypothesis	Tableau 6 – Climate Zone
Collective dwelling	Dwelling Type	Climate Type
Floor Investment	Carbon Price	Degrees of Reference
Ur	Discount Rate	
Price including Tax	Inflation Rate	
Individual dwelling	Insulation Limit	
Floor Investment		
Ur		
Price including Tax		

nomic Hypothesis » (Table 5), « Climate Zone » (Table 6), « Housing Characteristics » (Table 7), « Heating System » (Table 8), « Headcount » (Table 9) et « Headcount Sample ».

3.2.2 Présentation des tableaux

Parmi eux, les tableaux « Wall Investment », « Window Investment », « Roof Investment » et « Floor Investment » correspondent aux paramètres du mur, de la fenêtre, du toit et du sol du système d'isolation, y compris le coefficient de transfert de chaleur et son prix TTC. Toutes les données sont divisées en Collective Dwellings (comprenant CID et CCD) et Individual Dwellings selon les trois types de types de logements.

Le tableau « Housing Characteristics » répertorie tous les paramètres liés aux caractéristiques de la maison, tels que la période de construction de la maison, sa superficie totale, sa hauteur, le nombre d'étages, le nombre de maisons par étage et la proportion de surface vitrée. Toutes ces données peuvent être combinées à des formules de calcul spécifiques pour calculer la surface de chaque mur, fenêtre, toit et sol résidentiel, pouvant être utilisées ultérieurement pour calculer le coût de remplacement du système d'isolation thermique. Pour faciliter le calcul, ce modèle suppose que chaque maison est une figure carrée tridimensionnelle, de sorte que la surface du mur, de la fenêtre, du plafond et du sol puisse être obtenue par la surface totale. Il convient de noter qu'outre les paramètres susmentionnés relatifs à la zone résidentielle, le tableau fournit également les coefficients de transfert de chaleur (Um_0 , Uw_0 , Ur_0 et Uf_0) du système d'isolation

Tableau 7 – Housing Characteristics

Collective dwelling
Housing Type
Construction Period
Size Factor
Surface
Ceiling Height
Share of Windows
Dwelling per Floor
Floors
Intermittence
Um0
Uw0
Ur0
Uf0
Individual dwelling
Housing Type
Construction Period
Size Factor
Surface
Ceiling Height
Share of Windows
Dwelling per Floor
Floors
Intermittence
Um0
Uw0
Ur0
Uf0

Tableau 8 – Heating System

CID
Heating System
Efficiency
A Investment
B Investment
Maintenance Cost
Emission Factor
Energy Price
Same Switch Cost
Different Switch Cost
Delevery Type Cost
Specific Gas Price
Annual Variation
CCD
Heating System
Efficiency
A Investment
B Investment
Maintenance Cost
Emission Factor
Energy Price
Same Switch Cost
Different Switch Cost
Delevery Type Cost
Specific Gas Price
Annual Variation
ID
Heating System
Efficiency
A Investment
B Investment
Maintenance Cost
Emission Factor
Energy Price
Same Switch Cost
Different Switch Cost
Delevery Type Cost
Specific Gas Price
Annual Variation

Tableau 9 – Headcount

CID
Dwelling Type
Climate Zone
Construction Period
Size Factor
Heating System
Initial Age
Headcount
CCD
Dwelling Type
Climate Zone
Construction Period
Size Factor
Heating System
Initial Age
Headcount
ID
Dwelling Type
Climate Zone
Construction Period
Size Factor
Heating System
Initial Age
Headcount

thermique initial de la résidence. La raison pour laquelle il est répertorié séparément dans le vocabulaire plutôt que dans les quatre tableaux mentionnés ci-dessus est que ce paramètre varie en fonction du type, de la zone et de la période de construction de la maison. Et pour le système d'isolation thermique initial d'une maison, en raison de son coefficient de transfert de chaleur généralement élevé, il n'est pas possible de revenir au système d'isolation thermique initial. Le coût du remboursement après impôt correspondant n'est donc pas indiqué. Comme ci-dessus, toutes les données sont également divisées en Collective Dwellings (y compris CID et CCD) et Individual Dwellings selon les trois types de types de logements.

Les données du tableau « Heating System » ne concernent que le système de chauffage correspondant et la plupart d'entre eux sont liés à des formules économiques plus complexes, telles que le calcul du taux d'inflation, du taux d'amortissement, de la taxe sur les émissions de carbone, des coûts de maintenance et des factures d'énergie. Toutes les données sont divisées en trois catégories en fonction des trois principaux types de logements (CID, CCD, ID).

Les tableaux « Headcount » et « Headcount Sample » sont essentiellement les mêmes, sauf que le tableau « Headcount » présente les 32760 logements, alors que le tableau « Headcount Sample » ne représente qu'un échantillon de 1/1260 du tableau « Headcount ». Le tableau « Headcount Sample » est au total 26 logements choisis au hasard parmi le tableau « Headcount » et appartient à un type spécifique (trois types au total), à une zone spécifique (trois domaines au total), à une période de construction spécifique (un total de sept périodes de construction) et à une taille spécifique (trois au total taille) et à un âge initial spécifique du système de chauffage (total 20 ans). Ces 26 logements sont les données de remplacement que j'ai utilisées dans la première période pour vérifier l'exactitude du code. Ces deux tableaux sont étroitement liés à la présentation des résultats finaux, en particulier la colonne « Headcount », qui compte chaque type de maison dans la base de données. Nous pouvons également comprendre le tableau « Headcount » comme étant divisé en trois catégories par les trois principales catégories de logement.

Les données contenues dans les deux tableaux restants (« Climate Zone », « Economic Hypothesis ») ne sont pas affectées par le type de résidence, l'une étant liée à la région et l'autre au modèle dans son ensemble.

3.3 Implémentation de la programmation Python

3.3.1 Idées de développement

Au début de mon stage, je ne connaissais pas suffisamment tous les sujets et je n'avais aucune idée de la conception du programme, mon tuteur m'a demandé d'étudier un programme similaire en Python créé par une autre direction de l'entreprise, la direction Énergie-Environnement du CSTB, il y a quelques années. Le programme s'appelle « Dimosim » et est un programme écrit en Python 2.

En général, le programme utilise de nombreuses formules économiques complexes selon le principe de programmation orienté objet (POO), théoriquement, après l'instanciation, les résultats souhaités du calcul du système économique peuvent être obtenus en fonction de différents

paramètres. Malheureusement, le programme a été écrit il y a quelques années, la version que j'ai obtenue n'était pas complète et la plupart des personnes qui l'ont écrit ont quitté l'entreprise. Je n'ai donc pas réussi à exécuter le programme. De plus, je ne connais pas suffisamment les concepts de l'économie. Je ne peux donc comprendre que le concept de conception et le mode de fonctionnement du programme, qui n'est pas très complet.

Cependant, en étudiant « Dimosim » en détail, je m'en suis aussi inspiré. Par exemple, la conception de POO est adoptée et différents résultats de calcul sont obtenus en fonction de paramètres tels que le type, la taille et la période de construction de différentes maisons. Le but est de maintenir la lisibilité et l'uniformité de l'ensemble du programme. De plus, le nouveau programme doit également garantir que les opérations internes puissent filtrer les données et les formules en fonction des conditions, afin d'obtenir les résultats souhaités, quel que soit le type de conditions traitées.

3.3.2 Quelques mots sur la POO

Lorsque Python est utilisé pour concevoir des programmes de grande taille, son organisation interne devient relativement importante. La programmation orientée objet est la réponse actuelle à cette problématique et rend le développeur agile, c'est-à-dire rapidement réactif à des ajouts ou modifications du programme^[8].

La programmation orientée objet présente les avantages suivants :

- La classification des objets, encapsulant leurs données et méthodes appelables, facilite la gestion des fonctions, des variables et des données, et facilite l'appel de méthodes (réduction des paramètres répétés, etc.), en particulier lors de l'écriture de programmes volumineux.
- Avec la programmation orientée objet, vous pouvez manipuler les variables en tant qu'objets, ce qui rend les idées de programmation plus claires et concises et réduit l'occurrence de nombreuses variables redondantes.

3.3.3 Opération pratique

Avant d'écrire un programme pour ce modèle économique, j'ai écrit un petit programme permettant de convertir chaque table du fichier xlsx d'origine en un fichier csv via Pandas. Le but est de réduire le temps nécessaire au programme modèle pour lire les données. En règle générale, lorsque le même contenu et la même taille de fichier csv et de fichier xlsx sont lus via Pandas.read_csv et pandas.read_excel, le premier prend moins de temps que le second. Bien que la plupart des tableaux de ce xlsx ne soient pas volumineux, même convertis au format csv, ils ne permettent pas nécessairement de gagner beaucoup de temps, mais en tant que logiciel qui recherche la vitesse et l'efficacité des calculs, il semble plus raisonnable de convertir ces fichiers en fichiers csv et de les réutiliser. Une fois le fichier converti au format csv, tant que le fichier source n'est pas modifié, vous n'avez pas besoin d'exécuter à nouveau l'étape de conversion du fichier csv avant d'utiliser le programme suivant.

Lors de l'écriture du programme principal, j'ai d'abord écrit trois classes différentes basées sur

trois types de logements différents (CID, CCD, ID) pour localiser les données de système de chauffage pertinentes dans le tableau « Heating System ». Ensuite, en fonction de ces trois types de logements (j'ai écrit trois classes différentes pour représenter ces trois types de maisons. Du plus simple au plus complexe, j'ai tout d'abord créé trois classes pour calculer la valeur initiale en GHC du logement. Après cela, la fonction de remplacement du système de chauffage a été ajoutée à chaque logement et la valeur GHC après le remplacement du système a été obtenue avec succès. Tout au long du processus, j'ai combiné les caractéristiques du patrimoine génétique des POO, correspondant aux trois classes d'origine représentant le type logement, et dérivé trois nouvelles sous-classes ayant pour fonction de remplacer le système de chauffage. Après cela, la fonction de remplacement du système de chauffage a été ajoutée à chaque maison et la valeur GHC après le remplacement du système a été obtenue avec succès. Tout au long du processus, j'ai combiné les caractéristiques du patrimoine génétique des POO, correspondant aux trois classes d'origine représentant le type résidentiel, et dérivé trois nouvelles sous-classes ayant pour fonction de remplacer le système de chauffage.

À ce stade, les procédures importantes sont essentiellement terminées. Après cela, je peux non seulement calculer la valeur GHC d'origine selon la même méthode que le programme VBA d'origine, mais aussi trouver la plus petite valeur GHC en effectuant une comparaison les unes avec les autres dans l'hypothèse du remplacement du système d'isolation. De la même manière, je peux également comparer la valeur minimale de GHC avec la combinaison de chaque système de chauffage en supposant que le système d'isolation n'est pas remplacé.

Après avoir obtenu les valeurs ci-dessus, je peux disposer le système d'isolation thermique du logement (et/ou du système de chauffage) qui respecte le nombre maximal de systèmes d'isolation de remplacement (ou de systèmes de chauffage) en même temps, par ordre décroissant, du solde de GHC. Puis remplacez le système de chauffage du logement où le système d'isolation ne peut pas être remplacé, mais où il reste un surplus si le système de chauffage est remplacé. Après avoir remplacé le système de chauffage, l'âge initial du système de chauffage de la maison sera remis à zéro. Selon la même méthode, tous les logements avec ou sans système sont comparés l'année suivante.

3.3.4 Résultats obtenus

J'ai comparé un total de 26 logements de trois types de logements avec différents systèmes de chauffage dans la même zone climatique (H1), la période de construction (1), la taille (L) et la durée de vie du système de chauffage (0) (Figure 6 & 7). Le GHC minimum des trois types de maisons (CID, CCD, ID) est obtenu en comparant le GHC correspondant à 16 416 types, 14 592 types et 23040 types de systèmes d'isolation (selon les différentes permutations et combinaisons des quatre composants d'isolation thermique de mur, fenêtre, plafond et sol). Pour ces 26 logements, la limite supérieure du nombre total de logements avec systèmes d'isolation changés chaque année a été ajustée.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1		Climate Zone	Building Type	Construction Size	Factor	Heating Syst	Age Initial	Headcount	Initial GHC	Minimized G	Savings Ide	Insulation	Wall Investr	Windows Inv	Roof Investr	Floor Investr	Final GHC	Savings Fina	HS Switched	New Age of HS		
2	0 H1	CID		1 L	CI Electric He	0	6594	20391	20391	44286	28773	15513	True	4	0	2	5	28773	15113	ER	1	
3	1 H1	CID		1 L	CI Gas Boiler	0	5481	29043	23055	5988	5988	5988	False	0	0	0	0	29043	0	GB	1	
4	2 H1	CID		1 L	CI Condensin	0	6851	27040	22517	4523	4523	4523	False	0	0	0	0	27040	0	CGB	1	
5	3 H1	CID		1 L	CI Oil Boiler	0	38	3865	25207	8367	8367	8367	False	0	0	0	0	3865	33193	474 CGB	0	
6	4 H1	CID		1 L	CI Condensin	0	95	31778	24685	661	661	661	False	0	0	0	0	31778	0	OB	1	
7	5 H1	CID		1 L	CI Water So	0	19	25933	23023	2120	2120	2120	False	0	0	0	0	25933	0	EHP2	1	
8	6 H1	CID		1 L	CI Wood Sto	0	0	406798	30053	376745	Not Given	4	3	2	5	5	30053	376745	ER	0		
9	7 H1	CID		1 L	CI Wood Pel	0	0	400714	27282	373433	Not Given	4	0	2	0	0	27282	373433	GB	0		
10	8 H1	CID		1 L	CI Air Source	0	171	30375	24690	5688	5688	5688	False	0	0	0	0	30375	0	EHP	1	
11	9 H1	CCD		1 L	CC Gas Boile	0	5855	24936	20421	4515	4515	4515	False	0	0	0	0	24936	0	GB	1	
12	10 H1	CCD		1 L	CC Condensir	0	6216	22817	19675	3142	3142	3142	False	0	0	0	0	22817	0	CGB	1	
13	11 H1	CCD		1 L	CC Oil Boiler	0	302	3047	23136	7334	7334	7334	False	0	0	0	0	27979	2493	CGB	0	
14	12 H1	CCD		1 L	CC Condensir	0	0	2800	22321	5688	5688	5688	False	0	0	0	0	27979	26	CGB	0	
15	13 H1	CCD		1 L	CC Water So	0	0	566512	23925	54059	54059	54059	Not Given	4	0	2	0	23925	542585	OB	0	
16	14 H1	CCD		1 L	CC Wood Sto	0	0	564958	23923	54103	54103	54103	Not Given	4	0	2	0	23923	54123	GB	0	
17	15 H1	CCD		1 L	CC Wood Pel	0	158	20588	19187	3403	3403	3403	False	0	0	0	0	20588	0	WPB	1	
18	16 H1	CCD		1 L	CC Air Sourc	0	0	568163	23923	544240	544240	544240	Not Given	4	0	2	0	23923	544240	GB	0	
19	17 H1	ID		1 L	Electric Heat	0	5089	76954	48199	2875	2875	2875	True	4	0	5	0	48199	28757	EHP	0	
20	18 H1	ID		1 L	Gas Boiler (C	0	18317	47535	38786	8748	8748	8748	False	0	0	0	0	47535	0	GB	1	
21	19 H1	ID		1 L	Condensing C	0	22897	43236	36881	6445	6445	6445	False	0	0	0	0	43326	0	CGB	1	
22	20 H1	ID		1 L	Oil Boiler (O	0	2580	55731	43252	12479	12479	12479	True	4	0	5	0	43252	12479	OB	1	
23	21 H1	ID		1 L	Condensing C	0	6450	50724	40979	973	973	973	False	0	0	0	0	49213	1501	WPB	0	
24	22 H1	ID		1 L	Water Sourc	0	299	38640	35022	3674	3674	3674	False	0	0	0	0	38640	0	EHP2	1	
25	23 H1	ID		1 L	Wood Stone	0	4798	54189	42916	11273	11273	11273	True	4	0	5	0	42916	11273	WB	1	
26	24 H1	ID		1 L	Wood Pellet	0	1391	36581	33876	2705	2705	2705	False	0	0	0	0	36581	0	WPB	1	
27	25 H1	ID		1 L	Air Source El	0	2737	48704	40369	8335	8335	8335	True	4	0	5	0	40369	8335	EHP	1	

Figure 6 – Résultats du test de première année

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1		Climate Zone	Building Type	Construction Size	Factor	Heating Syst	Age Initial	Headcount	Initial GHC	Minimized G	Savings Ide	Insulation	Wall Investr	Windows Inv	Roof Investr	Floor Investr	Final GHC	Savings Fina	HS Switched	New Age of HS		
2	0 H1	CID		1 L	CI Electric He	1	6594	20391	20391	44286	28773	15513	False	4	0	2	5	20391	0	ER	1	
3	1 H1	CID		1 L	CI Gas Boiler	1	5481	29180	23173	6007	6007	6007	False	0	0	0	0	29180	0	GB	2	
4	2 H1	CID		1 L	CI Condensin	1	6851	27204	22517	4544	4544	4544	False	0	0	0	0	27204	0	CGB	2	
5	3 H1	CID		1 L	CI Oil Boiler	0	38	3865	25207	4523	4523	4523	False	0	0	0	0	27040	0	CGB	1	
6	4 H1	CID		1 L	CI Condensin	1	95	31248	24690	6625	6625	6625	True	4	0	2	0	24799	6629	COB	2	
7	5 H1	CID		1 L	CI Water So	1	19	26261	24141	2120	2120	2120	False	0	0	0	0	26261	0	EHP2	1	
8	6 H1	CID		1 L	CI Wood Pel	0	0	16763	16763	16763	16763	16763	False	4	0	5	0	16763	0	GB	1	
9	7 H1	CID		1 L	CI Wood Pel	0	0	16763	16763	16763	16763	16763	False	4	0	2	0	16763	0	GB	1	
10	8 H1	CID		1 L	CI Air Source	1	171	30603	24884	5725	5725	5725	False	0	0	0	0	30603	0	EHP	2	
11	9 H1	CCD		1 L	CC Gas Boile	1	5855	25046	20516	4530	4530	4530	False	0	0	0	0	25046	0	GB	2	
12	10 H1	CCD		1 L	CC Condensir	1	6216	22946	19789	3157	3157	3157	False	0	0	0	0	22946	0	CGB	2	
13	11 H1	CCD		1 L	CC Oil Boiler	0	302	22817	19675	3142	3142	3142	False	0	0	0	0	22817	0	CGB	1	
14	12 H1	CCD		1 L	CC Condensir	0	0	22817	19675	3142	3142	3142	False	0	0	0	0	22817	0	CGB	1	
15	13 H1	CCD		1 L	CC Water So	0	0	14130	14130	3142	3142	3142	False	4	0	2	0	14130	0	GB	1	
16	14 H1	CCD		1 L	CC Wood Sto	0	0	14130	14130	3142	3142	3142	False	0	0	0	0	20379	0	WPB	2	
17	15 H1	ID		1 L	Electric Heat	0	5089	29575	29575	4544	4544	4544	False	4	0	5	0	29575	0	EHP	1	
18	16 H1	ID		1 L	Gas Boiler (C	1	18317	47699	38921	8778	8778	8778	True	4	0	5	0	38921	8778	GB	2	
19	17 H1	ID		1 L	Condensing C	1	22897	43518	37042	6476	6476	6476	False	0	0	0	0	43518	0	CGB	2	
20	18 H1	ID		1 L	Oil Boiler (O	1	2580	32805	32808	2888	2888	2888	False	4	0	5	0	32808	0	OB	2	
21	19 H1	ID		1 L	Condensing C	0	6450	36581	33876	2705	2705	2705	False	0	0	0	0	36581	0	WPB	1	
22	20 H1	ID		1 L	Water Sourc	1	299	38868	35980	2888	2888	2888	False	0	0	0	0	38868	0	EHP2	2	
23	21 H1	ID		1 L	Wood Stone	1	4798	32507	32507	3142	3142	3142	False	4	0	5	0	32507	0	WB	2	
24	22 H1	ID		1 L	Wood Pellet	1	1391	36913	34197	2716	2716	2716	False	0	0	0	0	36913	0	WPB	2	
25	23 H1	ID		1 L	Air Source El	1	2737	48704	40369	8335	8335	8335	True	4	0	5	0	40369	8335	EHP	2	

Figure 7 – Résultats des tests pour la deuxième année

3.3.5 Défauts de conception

Si nous voulons trouver la valeur GHC d'une maison avec une condition spécifiée séparément selon le programme que j'ai écrit, c'est tout à fait réalisable et pratique. Mais si vous souhaitez trouver la valeur GHC dans des lots volumineux, sa mise en œuvre par le biais de mon programme prendra trop de temps.

Bien que j'ai obtenu correctement les résultats de calcul de 26 types de logements via la programmation Python, le temps nécessaire à l'ordinateur pour calculer les résultats pour chaque année est d'environ 10 minutes. Et je dois faire des prévisions pour 32 760 maisons au total. Si je suis cette procédure, je ne peux pas obtenir les résultats rapidement, même si cela est beaucoup plus long que le temps de calcul du programme original de VBA (environ une à deux heures). Par conséquent, ce programme est inacceptable. Je devrais commencer à chercher des moyens d'optimiser le programme.

3.4 Implémentation de la programmation Python avec la vectorisation

Parce que les programmes que j'ai écrits auparavant ne sont pas bons en termes d'efficacité et de rapidité de calcul, et que je n'ai pas pu trouver de solution à ce problème, je suis coincé dans une période de confusion.

Après une période de collecte de données et de discussions avec des experts en programmation Python d'autres sociétés, j'ai progressivement trouvé l'aube de l'espoir et me suis concentré sur la vectorisation.

Sous l'avis d'experts, je n'ai pas utilisé NumPy pour vectoriser directement, mais plutôt des Pandas pour utiliser indirectement NumPy pour la vectorisation. Cela peut perdre une petite partie de l'efficacité informatique, mais il est plus facile de trouver des erreurs de correction d'erreur en observant les données plus intuitivement. Bien que, dans la précédente étude, Pandas ne me soient pas inconnus, mais c'est la première fois que j'implémente la programmation vectorisée en Python à travers Pandas.

3.4.1 Quelques mots sur la vectorisation

Les vecteurs, les matrices et les tableaux de grandes dimensions sont des outils essentiels pour les calculs numériques. Lorsque des calculs répétés doivent être effectués sur un ensemble de valeurs d'entrée, il est naturel et avantageux de représenter les données sous forme de tableau et de représenter les calculs en fonction d'opérations sur les tableaux. Les calculs formulés de cette manière s'appellent vectorisation. En effectuant des opérations par lots sur des données de tableau, les calculs vectorisés ne nécessitent pas beaucoup de boucles explicites sur les éléments du tableau. Le résultat est concis et facile à gérer, et la mise en œuvre des opérations sur les baies peut être déléguée à des bibliothèques de bas niveau plus efficaces. Par conséquent, les calculs vectorisés sont beaucoup plus rapides que les calculs séquentiels par élément. Ceci est particulièrement important dans les langages interprétés, tels que Python, où le fait de parcourir en boucle les éléments d'un tableau, élément par élément, entraîne une surcharge de performances considérable^[9].

3.4.2 Retraitements des formules

La vectorisation du code signifie que le problème que nous résolvons est essentiellement vectorisé et ne nécessite que quelques astuces pour le rendre plus rapide. Bien sûr, cela ne signifie pas que ce soit simple ou direct, mais au moins cela n'a pas besoin de reconsidérer complètement notre problème^[10].

Par conséquent, avant la vectorisation du code, j'estime qu'il est nécessaire de retraitre les formules de calcul utilisées précédemment et de constater qu'il peut être simplifié ou omis. Avant de commencer à réécrire le code vectorisé, je passais beaucoup de temps à optimiser les formules mathématiques précédentes et à en discuter à plusieurs reprises avec le professeur pour certaines formules.

Grâce à une étude minutieuse de ces formules, non seulement j'ai réussi à réintégrer certaines de ces formules, mais j'ai également éliminé de nombreuses formules transitoires, mais peu utiles. Non seulement cela, j'ai également ajouté différentes couleurs spécifiques lors de la réécriture de la formule pour distinguer et marquer (Figure 8 & 9). Pratique pour une utilisation ultérieure lors de la rédaction de programmes. La méthode de différenciation des couleurs spécifique est la suivante :

$$\begin{aligned}
 D_{GHC_I} &= \frac{D_{investment_cost} \times HS_{inflation_rate_1st}}{HS_{discount_1st}} + \frac{D_{investment_cost} \times HS_{inflation_rate_2nd}}{HS_{discount_2nd}} \\
 &\quad + \frac{D_{uen_per_dwelling}}{HS_{efficiency}} \times HS_{energy_bill} + HS_{maintenance} + D_{insulation_cost} \\
 &= (HS_{a_investment} + HS_{b_investment} \times D_{uen_per_dwelling}) \\
 &\quad \times \frac{HS_{inflation_rate_1st} \times HS_{discount_2nd} + HS_{inflation_rate_2nd} \times HS_{discount_1st}}{HS_{discount_1st} \times HS_{discount_2nd}} \\
 &\quad + D_{uen_per_dwelling} \times \frac{HS_{energy_bill}}{HS_{efficiency}} + HS_{maintenance} + D_{insulation_cost} \\
 &= HS_{ghc_uen_condition_1} + D_{unvariable_cost_1} \tag{22}
 \end{aligned}$$

- **Where:**

$$D_{investment_cost} = HS_{a_investment} + HS_{b_investment} \times D_{uen_per_dwelling} \tag{23}$$

Figure 8 – Exemple 1 pour le retraitement des formules

- Le noir, la formule d'origine, mais ne sera pas utilisé dans les programmes suivants.
- Le vert, les données originales du programme seront utilisés sous forme de colonnes dans les programmes suivants.
- Le rouge foncé, une nouvelle colonne pouvant être calculée à partir de plusieurs colonnes, n'est pas itératif et sera utilisé dans les programmes suivants.
- Le rouge, plusieurs nouvelles colonnes pouvant être itérées à l'aide de calculs de plu-

sieurs colonnes et utilisées dans les programmes suivants.

- Le doré, soit une table composée de plusieurs colonnes itérées, soit une nouvelle table dérivée de la précédente. Cette liste sera utilisée dans les programmes suivants, et la génération de cette liste occupe la plus grande partie des ressources informatiques de l'ensemble du programme.
- L'orange, qui représente l'année du test en cours, compris entre 1 et 20, et sera utilisé dans le programme.
- Le bleu, il ne représente pas les données calculées par les procédures suivantes, mais les conditions à remplir pour pouvoir utiliser la formule spécifiée lors du remplacement du système de chauffage.

$$D_{unvariable_cost_1} = D_{insulation_cost} + HS_{fixed_cost_1} \quad (33)$$

if $HS_{to_switch} \in \{ER, GB, CGB, OB, COB, EHP2, WB, WPB, EHP\}$
⇒ $HS_{initial} \in \{ER, GB, CGB, OB, COB, EHP2, WB, WPB, EHP\}$
and $HS_{to_switch} = HS_{initial}$

$$D_{insulation_cost} = D_{insulation_cost_m_w} + D_{insulation_cost_r_f} \quad (34)$$

$$\begin{aligned} D_{insulation_cost_m_w} &= \frac{B_{surface_m}}{B_{dwelling_total}} \times D_{price_including_tax_m} \\ &+ \frac{B_{surface_w}}{B_{dwelling_total}} \times D_{price_including_tax_w} \end{aligned} \quad (35)$$

$$\begin{aligned} D_{insulation_cost_r_f} &= \frac{D_{surface}}{B_{floors} + 1} \times D_{price_including_tax_r} \\ &+ \frac{D_{surface}}{B_{floors} + 1} \times D_{price_including_tax_f} \end{aligned} \quad (36)$$

$$HS_{fixed_cost_1} = HS_{a_investment} \times HS_{time_factor_1} + HS_{maintenance} \quad (37)$$

$$HS_{maintenance} = \sum_{i=0}^{39} \frac{HS_{maintenance_cost} \times (1 + HS_{inflation_rate})^{i+year-1}}{(1 + D_{discount_rate})^i}, \quad year \in [1, 20] \quad (38)$$

Figure 9 – Exemple 2 pour le retraitement des formules

3.4.3 Intégration des données d'original

La clé de l'intégration des données est l'utilisation d'`itertools.product()` qui produit cartésien d'entrée iterables^[11].

En prenant comme exemple le logement type CID, je vais d'abord diviser les mots-clés liés à la zone climatique (3 au total), les mots-clés liés à la période de construction (7 au total), les mots-clés associés à la taille (3 au total), les mots-clés liés aux systèmes de chauffage (9 au total) et les mots liés à l'année d'utilisation du système de chauffage (20 au total) génèrent 5 listes indépendantes. Ensuite, j'ai fusionné et combiné `itertools.product()` pour générer une

table 11340×50 (Listing 1).

Une fois le tableau généré, je peux, avec pandas.merge, combiner les données restantes du tableau contenant ces mots-clés dans différentes colonnes basées sur ces mots-clés.

Quant aux données qui ne contiennent pas de mots-clés manquants, mais sont nécessaires dans le tableau, elles peuvent être directement fusionnées via pandas.concat.

```

1 class CID_Type(All_Types):
2     def __init__(self, year, *args):
3         super().__init__(year, *args)
4
5     def Combine_Lists(self):
6         self.list_combined = []
7         list_dwelling_type = ['CID']
8         list_climate_type = self.df_1.iloc[:, 0].tolist()
9         list_construction_period = self.df_2.iloc[:21, 1].tolist()[:3]
10        list_size_factor = self.df_2.iloc[:3, 2].tolist()
11        list_heating_system = [
12            'ER', 'GB', 'CGB', 'OB', 'COB', 'EHP2', 'WB', 'WPB', 'EHP'
13        ]
14        list_initial_age = list(range(20))
15        self.list_combined.append(list_dwelling_type)
16        self.list_combined.append(list_climate_type)
17        self.list_combined.append(list_construction_period)
18        self.list_combined.append(list_size_factor)
19        self.list_combined.append(list_heating_system)
20        self.list_combined.append(list_initial_age)
21        return self.list_combined
22
23    def Generate_DF(self):
24        self.Combine_Lists()
25        itertools_product = list(itertools.product(*self.list_combined))
26        self.df = pd.DataFrame(itertools_product,
27                               columns=[
28                                   'Dwelling Type', 'Climate Type',
29                                   'Construction Period', 'Size Factor',
30                                   'Heating System', 'Initial Age'
31                               ])
32        return self.df
33
34    def Merge_DFs(self):
35        self.Generate_DF()
36        self.df = self.df.merge(self.df_1, how='left', on=['Climate Type'])
37        self.df = self.df.merge(self.df_2.iloc[:21, 1:9],
38                               how='left',
39                               on=['Construction Period', 'Size Factor'])
40        return self.df
41
42    def Add_Columns(self):
43        self.Merge_DFs()
44        self.df[self.df_5.columns[4]] = self.df_5.iloc[0, 4]
45        return self.df

```

```

46
47 def Concat_DFs(self):
48     self.Add_Columns()
49     self.df = pd.concat([self.df, self.df_3.iloc[0:11340, 6]], 
50     axis=1,
51     join='outer')
52     return self.df

```

Listing 1 – Exemple d'intégration de données

3.4.4 Structure du programme

Lors de l'écriture de programmes vectorisés, j'utilisais toujours la programmation POO, dans le but de réduire l'occupation excessive de la mémoire de l'ordinateur au cours des différentes étapes du programme (Figure 10). Cependant, le nouveau programme est assez différent de la conception précédente. La conception de ce programme vectorisé a été principalement inspirée lors de la réécriture de la formule. Les idées principales sont les suivantes :

- Fusionner les données pour générer une table contenant toutes les données ;
- Générer des colonnes pouvant être calculées sans itération ;
- Par itération, générez les colonnes pouvant être itérées par calcul ;
- Toutes les colonnes itérables générées sous forme de table pour participer au calcul ;
- Selon les conditions correspondantes, différentes formules sont automatiquement sélectionnées dans le calcul.

Dans la figure 10, toutes les petites cases bleues sont parentales et le sens de la flèche indique le sens de l'héritage.

Toutes les petites cases vertes sont les classes qui doivent être générées lors de la fusion des données initiales. Les trois types de logements sont divisés en deux branches, car les informations sur le système d'isolation thermique utilisé incluent le coefficient de transfert de chaleur et le coût après impôt.

Toutes les petites cases mauves sont les classes qui produisent des colonnes calculables (y compris des colonnes itérables ou non). Toutes les colonnes itérables sont liées à la conductivité thermique et au coût après impôt du système d'isolation.

Tous les petits carrés rouges représentent des classes qui doivent être calculées de manière uniforme dans un tableau complet. Cette table est l'ensemble des colonnes itérables décrites ci-dessus. Selon la formule réécrite, deux petites boîtes rouges du même type de maison sont superposées pour obtenir la valeur finale de GHC. Pour les logements CID, CCD et ID, le nombre de colonnes dans ce tableau est 16416, 14592 et 23040, ce qui correspond au nombre de types de systèmes d'isolation possibles pour chaque type d'itération de logement. Le nombre de lignes pour ces trois types de logements est 11340, 10080 et 11340, ce qui correspond au nombre de types de chaque type de logement. Chaque colonne représente chaque type de résidence. Ce que nous devons ensuite faire, c'est calculer la valeur GHC de chaque maison

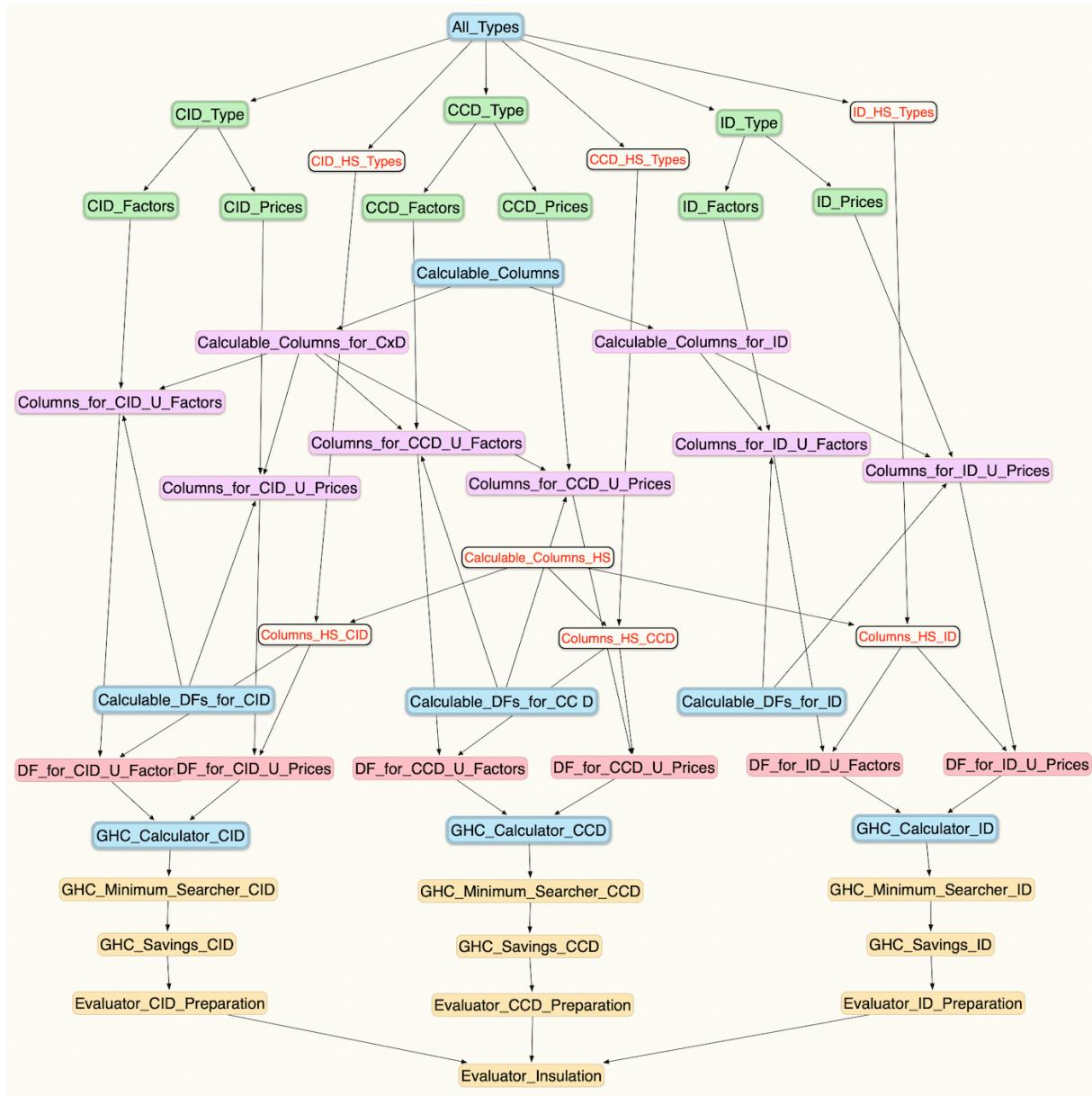


Figure 10 – Structure de classe du programme

après avoir supposé que chaque type de système d'isolation est remplacé et obtenir la plus petite valeur GHC, qui est la valeur minimale de chaque ligne du tableau.

Toutes les petites cases jaunes représentent la classe d'étapes à suivre pour évaluer si le système d'isolation doit être remplacé après le calcul de la valeur GHC.

Et toutes les petites cases blanches ne représentent que la classe de calcul qui n'a rien à voir avec le logement, mais uniquement avec le système de chauffage.

3.4.5 Programmation vectorisée comme un puzzle

Comme ce que j'ai mentionné précédemment, lors du calcul d'un tableau dans son ensemble, il se heurtera inévitablement à des conditions de calcul différentes. Si c'était mon premier programme, il pourrait être traité avec une simple déclaration « if ... else... ». Mais si vous avez affaire à une table entière, vous avez besoin d'un petit truc.

Pour notre modèle, nous allons rencontrer différents choix de conditions lors de la prévision de remplacement du système de chauffage. Au total, trois types principaux de logements peuvent être divisés en deux catégories : en utilisant des systèmes de chauffage indépendants (CID et ID) ou en utilisant un système de chauffage commun (CCD).

En termes simples, s'agissant du problème de remplacement du système de chauffage des logements de chauffage indépendants (CID et IC) ou des logements de système de chauffage commun (CCD), le tableau constitué des colonnes itératives mentionnées ci-dessus peut être imaginé comme un tableau 9×9 (Figure 11) ou 8×8 (Figure 12) tableau, dont 9 ou 8 correspond au type de système de chauffage que le logement peut avoir. La rangée horizontale du tableau correspond au nom du système de chauffage à remplacer et la rangée verticale au nom du système de chauffage initial.

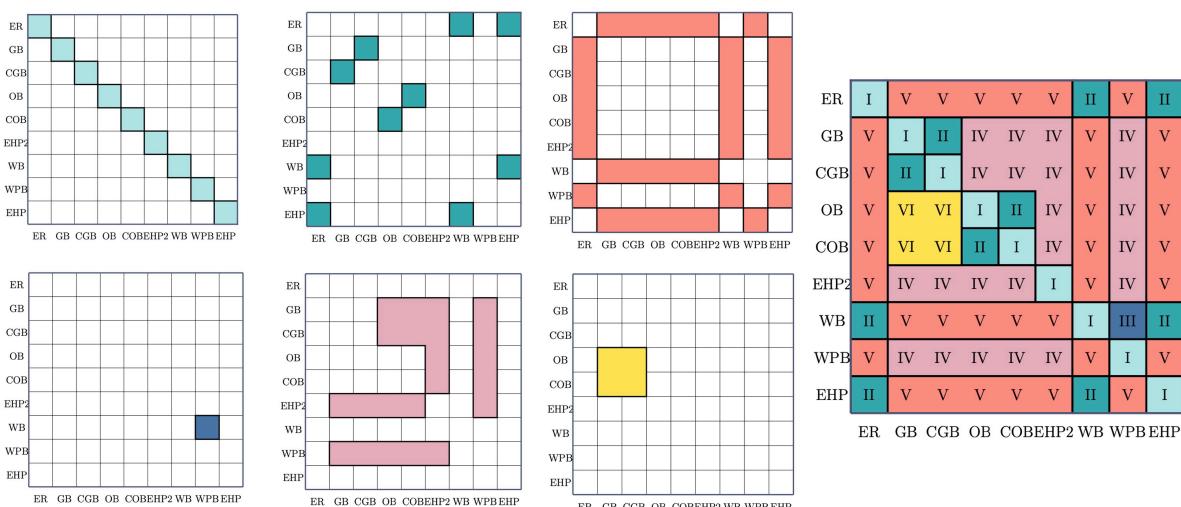


Figure 11 – Conditions des logements de chauffage indépendant (CID et IC)

Comme la montre par les figures, mon modèle a résumé six conditions différentes et ne se chevauchant pas lorsqu'il s'agit de remplacer un système de chauffage. En combinant ces six

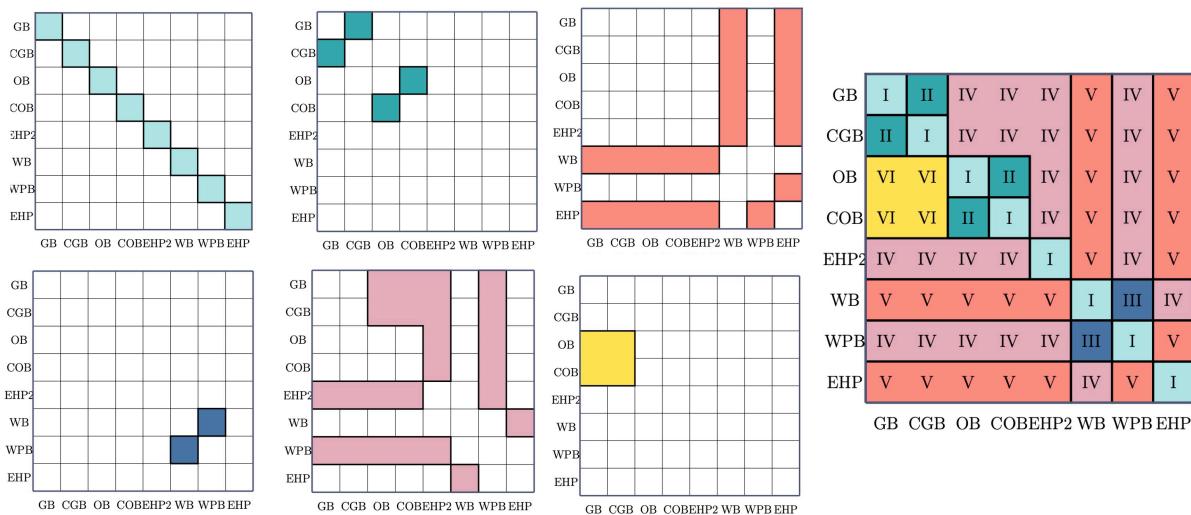


Figure 12 – Conditions des logements de système de chauffage commun (CCD)

conditions, il est possible d'obtenir un tableau 9×9 ou 8×8 complet et ne se chevauchant pas.

Bien entendu, le processus réel ne peut être aussi simple et intuitif que les 9×9 ou 8×8 carrés. La méthode que j'ai utilisée consistait à trier la table entière horizontalement selon le type de système de chauffage cible, puis à la diviser en fonction du type de système de chauffage. La forme originale est divisée en 9 ou 8 petites formes. Ces petites tables sont les mêmes, sauf que les systèmes de chauffage cibles correspondants sont différents et que la correspondance un à un résulte de l'itération du système d'isolation. En effet, ces petites tables sont en réalité une colonne dans la figure.

Les avantages de cette approche sont doubles :

- Réduction de la taille déguisée, simplifiant la difficulté d'établissement de condition.
- Pendant le calcul, le traitement par lots des tables réduites peut réduire l'utilisation de la mémoire en même temps et accélérer le calcul.

3.4.6 Effets opérationnels et recommandations futures

Grâce au traitement de la programmation vectorisée, la vitesse d'exécution finale du programme peut atteindre environ 10 minutes par année de test (testée sur un ordinateur disposant d'une mémoire importante). En raison du nombre considérable de calculs effectués à chaque fois, il a été constaté lors de l'examen du logiciel complet au cours des années qu'il nécessitait 4-5Go de mémoire. Pour un ordinateur général, la vitesse du programme varie en fonction de la mémoire ou du processeur de l'ordinateur.

En raison de la durée du stage, je n'ai pas eu la possibilité de combiner des méthodes parallèles et vectorisées pour augmenter la vitesse des opérations.

En fait, j'ai envisagé d'utiliser Dask ou Swifter pour remplacer la fonction apply utilisée par Pandas dans le code source. Cette fonction est le processus de génération d'un tableau complet

par l'itération de liste, c'est également la partie la plus gênante du programme. Au début, si vous vouliez parcourir directement des dizaines de milliers de colonnes, cela prenait une heure (plus tard, je réduisais leur taille de 8 à 9 fois en utilisant leurs caractéristiques communes). Mais ensuite, j'ai découvert que la fonction `apply` de Dask ne pouvait être utilisée qu'avec `axis = 1`^[12], ce qui entraînait en conflit avec mon programme. J'ai donc dû abandonner.

Pour le moment, Dask ne prend en charge que `axis = 1`, et `swifter` est donc limité à `axis = 1` pour les grands ensembles de données lorsque la fonction ne peut pas être vectorisée^[13].

Je pense que si nous voulons augmenter encore la vitesse, nous pouvons envisager d'utiliser la fonction de threading de Python^[14]. Toutefois, dans le processus de fonctionnement parallèle vectorisé du programme, la demande de mémoire informatique ne peut qu'augmenter. Pour ce faire, il faut un ordinateur avec plus de mémoire, ce que je ne peux pas faire avec mon ordinateur personnel.

4 Conclusion

Ce stage a été une chance pour moi, après plusieurs mois de la recherche afin que je puisse commencer mon premier stage formel dans le domaine d'informatique en France. Heureusement, j'ai pu être admis par cette offre de CSTB, et passer une inoubliable durée de 6 mois avec une bonne équipe.

Le contenu de mon stage est le développement d'un modèle de simulation économique complexe sur la plate-forme Python. Tout au long du processus, j'ai traité beaucoup de problèmes rencontrés en consultant d'autres experts, en pensant de manière autonome pour trouver des réponses, en m'apprenant à moi-même et en ajoutant de la valeur. J'ai réalisé de nombreux défis et percées et reçu une grande satisfaction.

En plus d'améliorer et de renforcer mon niveau technique, ma compréhension et ma réflexion sur la programmation Python ont également été exercées. Pour une personne qui n'est pas majeure en informatique au niveau du premier cycle, ce stage a plus ou moins compensé nombre de mes lacunes et carences dans l'expérience.

Après six mois de stage, j'ai également commencé à repenser mes objectifs de développement futurs. Au début, j'ai choisi de venir en France pour l'informatique principalement parce que j'étais optimiste quant à la future application et à l'importance de la science des données, mais j'avais peu d'idées d'optimisation ou d'algorithmes. Mais après six mois de développement logiciel, en particulier concernant l'optimisation et la vectorisation de programmes, je me suis également intéressé aux structures de données et aux algorithmes. Si j'en ai l'occasion, je pense qu'il serait intéressant de poursuivre mes recherches et mes études dans ce domaine. Parallèlement, au cours de mon prochain stage, j'ai eu l'idée de participer à des tâches plus complexes et plus difficiles, telles que l'apprentissage automatique et même l'intelligence artificielle, une idée que je n'osais pas avoir auparavant.

J'espère que dans la longue période pour trouver un stage, je pourrai mieux compenser mes lacunes, afin de pouvoir avoir de la chance lorsque je rencontrerai le stage idéal mentionné ci-dessus.

Glossaire

CCD : Collective Dwellings with Collective heating system

CID : Collective Dwellings with Individual heating system

CSTB : Centre Scientifique et Technique du Bâtiment

DA2E : Direction Analyse et Études économiques

GHC : Global Heating Costs

ID : Individual Dwellings

POO : Programmation orientée Objet

UEN : Useful Energy Needs

Ug : the overall heat transfer coefficient

Références

- [1] De Perthuis, C. and P.-A. Jouvet (2013). Le Capital Vert. Odile Jacob
- [2] Chevalier, J.-M., M. Cruciani, and P. Geoffron (2013). Transition Énergétique, Les vrais choix. Odile Jacob
- [3] EC (2014). European union energy in figures - pocketbook 2014. European Commission
- [4] Charlier, D. and A. Risch (2012, July). Evaluation of the impact of environmental public policy measures on energy consumption and greenhouse gas emissions in the french residential sector. Energy Policy 46, 170 – 184
- [5] Jochem, E. et al. (2000). Energy end-use efficiency. Technical report, UNDP/UNDESA/WEC: Energy and the Challenge of Sustainability. World Energy Assessment. New York: UNDP, 173 – 217
- [6] Li, D. H., L. Yang, and J. C. Lam (2013). Zero energy buildings and sustainable development implications-a review. Energy 54, 1 – 10
- [7] El Beze J. (2018, December). Les rôles de la substitution et de l'efficacité énergétiques dans la décarbonation du parc de logements en France, thèse de doctorat, Université Paris-Dauphine-PSL
- [8] Ziadé T.(2009). Programmation Python : Conception et optimisation. Eyrolles.
- [9] Johansson R.(2019). Numerical Python, https://doi.org/10.1007/978-1-4842-4246-9_2
- [10] Nicolas P.-R. (2017, May). From Python to Numpy, <http://www.labri.fr/perso/nrougier/from>
- [11] <https://docs.python.org/3.7/library/itertools.html#itertools.product>
- [12] <https://docs.dask.org/en/latest/dataframe-api.html#dask.dataframe.DataFrame.apply>
- [13] <https://github.com/jmcarpenter2/swifter/blob/master/docs/documentation.md>
- [14] <https://docs.python.org/3/library/threading.html>