

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ  
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE MÉCANIQUE  
M. Ing

PAR  
RENÉ BELLEMARE

DÉVELOPPEMENT ET INTÉGRATION DE NOUVEAUX MODULES DE  
SIMULATION AU LOGICIEL D'ANALYSE ÉNERGÉTIQUE DES BÂTIMENTS

MONTRÉAL, 26 MAI 2003

© droits réservés René Bellemare

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Louis Lamarche, président du jury  
Département de génie mécanique à l'École de Technologie Supérieure

M. Stanislaw Kajl, directeur de mémoire  
Département de génie mécanique à l'École de Technologie Supérieure

M. Marc-Antoine Roberge, directeur du département Design/Énergie  
Dupras Ledoux ingénieurs

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT JURY ET PUBLIC

LE 23 AVRIL 2003

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

# **DÉVELOPPEMENT ET INTÉGRATION DE NOUVEAUX MODULES DE SIMULATION AU LOGICIEL D'ANALYSE ÉNERGÉTIQUE DES BÂTIMENTS**

René Bellemare

## RÉSUMÉ (150 mots)

Premièrement, on démontre dans ce mémoire les capacités de modélisation d'un système de traitement de l'air d'un bâtiment à l'aide du logiciel EnergyPlus. La validation du modèle programmé dans le logiciel EnergyPlus a été réalisé à l'aide d'une base de données enregistrées. On y présente également l'analyse comparative entre les résultats de simulation obtenus à partir d'un logiciel de référence (DOE-2E) et ceux obtenus avec EnergyPlus. Les résultats de cette analyse permettent de remarquer de légères différences entre ceux-ci. Deuxièmement, on présente la méthodologie utilisée lors de l'élaboration et de l'intégration d'un module de simulation dans le logiciel EnergyPlus. Le module de simulation développé dans ce volet permet de simuler l'opération d'une chaudière en fonction des principes de contrôle biénergie établis par Hydro-Québec. La fiabilité du nouveau module de simulation a été éprouvée avec succès. L'impact du pas de temps de calculs sur les coûts d'électricité y est également présenté.

# **DÉVELOPPEMENT ET INTÉGRATION DE NOUVEAUX MODULES DE SIMULATION AU LOGICIEL D'ANALYSE ÉNERGÉTIQUE DES BÂTIMENTS**

René Bellemare

## **SOMMAIRE**

Ce mémoire de maîtrise peut être divisé en deux volets distincts. Dans le premier volet, on démontre les capacités de modélisation d'un système de traitement de l'air d'un bâtiment à l'aide du logiciel EnergyPlus v.1 build23. Pour ce faire, une partie du bâtiment abritant l'École de Technologie Supérieure (ETS) a servi de modèle. Un des onze systèmes VAV et les zones qu'il dessert ont été simulés à l'aide des logiciels EnergyPlus et DOE-2E. La validation du modèle programmé dans le logiciel EnergyPlus a été réalisé à l'aide d'une base de données enregistrées sur 4 zones internes et sur le système CVCA qui les alimente. Les résultats de cette analyse comparative permettent d'affirmer qu'il est possible, à partir d'un relevé détaillé des zones d'un bâtiment, de simuler avec précision l'opération des systèmes CVCA et les zones qu'il dessert. On présente également dans ce volet l'analyse comparative entre les résultats de simulation obtenus à partir d'un logiciel de référence (DOE-2E) et ceux obtenus avec EnergyPlus. Les résultats de cette deuxième analyse permettent de remarquer de légères différences entre les résultats des deux logiciels. Finalement, on y commente les particularités et les observations faites lors de la programmation et de l'analyse des résultats. Certaines anomalies détectées lors de l'analyse des résultats sont également décrites dans ce volet. Dans le deuxième volet, on présente la méthodologie utilisée lors de l'élaboration et de l'intégration d'un module de simulation dans le logiciel EnergyPlus version 1.0.2.008. Le module de simulation développé dans ce volet permet de simuler l'opération d'une chaudière en fonction des principes de contrôle biénergie établis par Hydro-Québec. Afin de le valider, un modèle fictif d'un édifice à bureaux a été programmé sous la version 1.0.2.008 d'EnergyPlus. L'édifice a été programmé en respectant les différentes lois et normes en matière d'énergie. Le modèle ainsi créé a permis d'éprouver, avec succès, la fiabilité du nouveau module de simulation. Pour finir, on y présente une analyse comparative des coûts énergétiques entre trois modèles de simulation, soit un modèle équipé d'une chaudière électrique/gaz naturel et d'un contrôle biénergie, un second équipé d'une chaudière électrique et un troisième muni d'une chaudière au gaz naturel. L'impact du pas de temps de calculs sur les coûts d'électricité y est également présenté.

# **DEVELOPMENT AND INTEGRATION OF NEW SIMULATION MODULES TO ANALYZE ENERGY BUILDING SOFTWARE**

René Bellemare

## **ABSTRACT**

This master project can be divided into two distinct parts. In the first part, we demonstrate the modelling capabilities of a building's HVAC system using EnergyPlus v.1 build 23 software. A section of the École de Technologie Supérieure (ÉTS) was used as a model. Of the eleven VAV systems and associated zones, one has been simulated using both EnergyPlus and DOE-2E software. A database created from four internal areas and their associated HVAC system served to validating the EnergyPlus software programmed model. Results from this comparative analysis confirm that it is possible to precisely simulate the operation of HVAC systems and their associated areas based on a detailed overview of building's areas. Moreover, we show the comparative analysis between simulation results obtained from the reference software (DOE-2E) and those obtained from EnergyPlus software. Those results also highlight differences between both software results. Finally, we explain characteristics and observations that were made during the modelling and results analysis.

In the second part, we present the methodology used for the creation and the integration of the simulation module into EnergyPlus version 1.0.2.008 software. The simulation module created in this part can be used to simulate the operation of a boiler according to biénergie controls established by Hydro-Québec. To help validating this module, a fictitious office building model has been programmed using EnergyPlus version 1.0.2.008. This building model was programmed in accordance with energy laws and standards. Thus, this model has successfully proved the reliability of the new simulation module. At the end, a comparative analysis of energy costs between three simulation models and the impact of the computing timestamp on electricity costs is also presented.

## **REMERCIEMENTS**

En premier lieu, je tiens à remercier mon directeur de mémoire M. Stanislaw Kajl pour m'avoir encouragé à poursuivre mes études à l'École de Technologie Supérieure et pour son soutien tout au long de mes études. Je remercie également M. Marc-Antoine Roberge pour m'avoir fait découvrir le domaine de la simulation énergétique sur les bâtiments.

Je voudrais également remercier le Fonds québécois de la recherche sur la nature et les technologies (NATEQ) et l'École de Technologie Supérieure pour les différentes bourses qu'ils m'ont octroyées.

Finalement, je tiens à remercier de tout cœur mes parents, mes amis et tout particulièrement mon épouse et ma fille pour leur soutien moral.

## TABLE DES MATIÈRES

	Page
SOMMAIRE .....	i
ABSTRACT .....	ii
REMERCIEMENTS .....	iii
TABLE DES MATIÈRES .....	iv
LISTE DES TABLEAUX.....	vi
LISTE DES FIGURES.....	viii
LISTE DES ABREVIATIONS ET SIGLES .....	ix
INTRODUCTION .....	1
CHAPITRE 1 REVUE DE LITTÉRATURE .....	3
1.1 Validation des algorithmes de calculs.....	4
1.2 Deux parties distinctes .....	5
CHAPITRE 2 Analyse DES CAPACITÉS du logiciel ENERGYPLUS.....	7
2.1 Objectifs .....	7
2.2 Méthodologie .....	8
2.3 Modélisation.....	8
2.4 Validation du modèle .....	12
2.5 Comparaison entre EnergyPlus et DOE-2.....	18
2.6 Observations et anomalies .....	22
2.6.1 Remarques et observations portant sur la modélisation.....	22
2.6.1.1 Saisie de données .....	22
2.6.1.2 Définition des parois .....	23
2.6.1.3 Outils de conversion.....	24
2.6.1.4 Agencement des systèmes mécaniques.....	25
2.6.1.5 Temps de calculs et précision des calculs .....	25
2.6.1.6 Contrôle des éléments mécaniques .....	27
2.6.2 Anomalies rencontrées .....	27
2.6.2.1 Pertes de débit .....	27
2.6.2.2 Respect des limites.....	27
CHAPITRE 3 Développement et intégration d'un module de simulation.....	29

3.1	Objectif.....	30
3.2	Méthodologie .....	30
3.3	Développement .....	31
3.3.1	Règles de contrôle .....	31
3.3.2	Contrôles existants .....	32
3.3.3	Modules existants.....	35
3.3.4	Nouveau module .....	37
3.3.4.1	Modifications et ajouts.....	38
3.3.4.2	Outils diagnostiques.....	41
3.4	Validation du module.....	44
3.4.1	Modèle de simulation.....	44
3.4.1.1	Bâtiment .....	44
3.4.1.2	Charges internes .....	47
3.4.1.3	Systèmes mécaniques.....	48
3.4.1.4	Séquence d'opération .....	51
3.4.2	Analyse des résultats.....	52
	CONCLUSION .....	56
	ANNEXES	
1 :	Module de simulation biénergie.....	58
2 :	Sommaire des résultats avec un pas de temps de calculs de 10 minutes .....	88
3 :	Sommaire des résultats avec un pas de temps de calculs de 60 minutes .....	94
4 :	Publication .....	100
	BIBLIOGRAPHIE .....	108

## LISTE DES TABLEAUX

	Page	
Tableau I	Caractéristiques thermodynamiques des composantes des murs extérieurs.....	45
Tableau II	Caractéristiques thermodynamiques des composantes du plancher.....	45
Tableau III	Caractéristiques thermodynamiques des composantes de la toiture .....	45
Tableau IV	Caractéristiques thermodynamiques des panneaux de verres .....	46
Tableau V	Horaires d'exploitation du bâtiment de 1h à 12h .....	47
Tableau VI	Horaires d'exploitation du bâtiment de 13h à 24h.....	47
Tableau VII	Influence du pas de temps de calculs sur le coût de l'électricité .....	55
Tableau VIII	Modèle #1 – Option 1 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 10 minutes .....	89
Tableau IX	Modèle #1 – Option 1 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 10 minutes.....	89
Tableau X	Modèle #1 – Option 1 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes .....	90
Tableau XI	Modèle #1 – Option 2 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 10 minutes .....	90
Tableau XII	Modèle #1 – Option 2 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 10 minutes.....	91
Tableau XIII	Modèle #1 – Option 2 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes .....	91
Tableau XIV	Modèle #2 : Coûts d'électricité avec un pas de temps de calculs de 10 minutes.....	92
Tableau XV	Modèle #3 : Coûts d'électricité avec un pas de temps de calculs de 10 minutes.....	92
Tableau XVI	Modèle #3 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes .....	93
Tableau XVII	Modèle #1 – Option 1 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 60 minutes .....	95
Tableau XVIII	Modèle #1 – Option 1 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 60 minutes.....	95
Tableau XIX	Modèle #1 – Option 1 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes .....	96
Tableau XX	Modèle #1 – Option 2 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 60 minutes .....	96
Tableau XXI	Modèle #1 – Option 2 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 60 minutes.....	97
Tableau XXII	Modèle #1 – Option 2 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes .....	97
Tableau XXIII	Modèle #2 : Coûts d'électricité avec un pas de temps de calculs de 60 minutes .....	98

Tableau XXIV	Modèle #3 : Coûts d'électricité avec un pas de temps de calculs de 60 minutes .....	98
Tableau XXV	Modèle #3 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes .....	99

## LISTE DES FIGURES

	Page	
Figure 1	Schéma d'un système CVCA à volume d'air variable .....	10
Figure 2	Schéma du système secondaire UTA-6 et du système primaire .....	11
Figure 3	Variation de la température d'alimentation du système UTA-6 le 18 juin 2001 .....	14
Figure 4	Variation de la température de la salle de cours 2304 le 18 juin 2001. ....	15
Figure 5	Variation du débit volumétrique d'air soufflé de la salle de cours 2304 le 18 juin 2001.....	16
Figure 6	Variation du débit volumétrique d'air du ventilateur de l'unité UTA-6 le 18 juin 2001.....	17
Figure 6	Influence du choix de l'algorithme de calcul du coefficient de convection interne sur la température ambiante du local 2304.....	19
Figure 7	Répartition journalière de la température ambiante de la salle de cours 2304.....	20
Figure 8	Répartition journalière du débit volumétrique d'air soufflé de la salle de cours 2304. ....	20
Figure 9	Répartition du débit volumétrique d'air du ventilateur du système UTA-6 pour les journées du lundi et mardi. ....	21
Figure 10	Répartition journalière de la demande de refroidissement au serpentin du système UTA-6 pour les journées du lundi et mardi.....	22
Figure 11	Partitions internes des zones desservies par le système UTA-6 et modélisées dans EnergyPlus .....	24
Figure 12	Climatisation indirecte des salles de toilettes .....	25
Figure 13	Variation du débit volumétrique d'air du système UTA-6 en fonction du pas de temps. ....	26
Figure 14	Variation de la demande de refroidissement du système UTA-6 en fonction du pas de temps.....	26
Figure 15	Structure interne du logiciel EnergyPlus – Niveau supérieur .....	36
Figure 16	Structure partielle du module « <i>ManageHVAC.f90</i> » .....	37
Figure 17	Zonage du bâtiment.....	46
Figure 18	Schéma du système secondaire et du système primaire.....	49
Figure 19	Fonctionnement du système biénergie.....	52
Figure 20	Coûts mensuels d'énergie .....	54

## LISTE DES ABREVIATIONS ET SIGLES

Con	Conductivité thermique, W/m-K
Cs	Chaleur spécifique, J/kg-K
E <sub>b</sub>	Émissivité hémisphérique de la surface interne de la vitre
E <sub>f</sub>	Émissivité hémisphérique de la surface externe de la vitre
R	Résistance thermique, m <sup>2</sup> -K/W
R <sub>bs</sub>	Coefficient de réflexion solaire à incidence normale à la surface interne de la vitre
R <sub>bv</sub>	Coefficient de réflexion visible à incidence normale à la surface interne de la vitre
R <sub>fs</sub>	Coefficient de réflexion solaire à incidence normale à la surface externe de la vitre
R <sub>fv</sub>	Coefficient de réflexion visible à incidence normale à la surface externe de la vitre
T <sub>h</sub>	Épaisseur de la vitre, m
T <sub>ir</sub>	Coefficient de transmissibilité à incidence normale à la surface de la vitre
T <sub>s</sub>	Coefficient de transmissibilité solaire à incidence normale à la surface de la vitre
T <sub>v</sub>	Coefficient de transmissibilité visible à incidence normale à la surface de la vitre
ρ	Masse volumétrique, kg/m <sup>3</sup>
ε	Émissivité d'une surface opaque
α <sub>s</sub>	Coefficient d'absorption solaire d'une surface opaque
α <sub>v</sub>	Coefficient d'absorption visible d'une surface opaque

## INTRODUCTION

Depuis les 20 dernières années, le rôle des outils de simulation dans le design et l'ingénierie des bâtiments a sans cesse grandi. L'utilisation des outils de simulation permet désormais d'optimiser le design des systèmes électromécaniques avec l'architecture du bâtiment. Il est maintenant possible d'évaluer de façon précise et relativement rapide l'impact de certaines décisions prises lors du design sur la consommation d'énergie et sur le dimensionnement des composantes électromécaniques du bâtiment. Le défi pour la prochaine décennie est sans aucun doute l'intégration complète et totale des outils de simulation dans les différentes étapes de design d'architecture et d'ingénierie des bâtiments.

Au cours des années, plusieurs logiciels de simulation des bâtiments ont vu le jour. Évidemment, la précision de ces logiciels varie en fonction de la méthode de calculs et des hypothèses simplificatrices utilisées. En avril 2001, un nouveau logiciel est venu s'ajouter à la liste des logiciels les plus performants, tels DOE-2E et Blast. Contrairement à ses prédecesseurs, ce nouveau logiciel, nommé EnergyPlus, permet de simuler toutes sortes d'arrangements de systèmes mécaniques du bâtiment. Un des points forts d'EnergyPlus par rapport à ses prédecesseurs, est l'intégration de tous les aspects de la simulation, soit les charges, les systèmes secondaires et les systèmes primaires. De cette façon, EnergyPlus calcule directement la réponse thermique du bâtiment suite à un sous-dimensionnement des systèmes secondaires et/ou primaires. Ainsi, il est possible désormais d'évaluer l'impact d'un sous-dimensionnement des équipements mécaniques sur le confort thermique des occupants d'un bâtiment. Il devenait donc intéressant pour moi de baser mon mémoire de maîtrise sur ce logiciel.

Mon mémoire est composé de trois chapitres. Le premier chapitre présente la revue de littérature portant sur la validation des algorithmes de calculs employés par EnergyPlus. On y retrouve également les raisons pour lesquelles mon mémoire est divisé en deux

parties distinctes. Les chapitres 2 et 3 traitent respectivement de l'analyse des capacités du logiciel EnergyPlus et du développement et de l'intégration d'un module de simulation. Plus précisément, le chapitre 2 présente la validation, à partir d'une base de données d'enregistrement, des résultats de simulation d'EnergyPlus, l'étude comparative des résultats d'EnergyPlus avec ceux de DOE-2E ainsi que les remarques et observations rencontrées lors de la modélisation et de l'analyse des résultats. Quant au chapitre 3, on y présente la méthodologie employée lors de l'élaboration d'un module de simulation, le développement et la validation de ce dernier et finalement l'analyse des résultats sous forme de tableaux et graphiques.

## **CHAPITRE 1**

### **REVUE DE LITTÉRATURE**

L'existence du logiciel EnergyPlus est directement liée à certaines imperfections de plus en plus évidentes de ses prédecesseurs : Blast et DOE-2. Les logiciels Blast et DOE-2, bien qu'encore les outils de simulations de référence et faisant partie des outils les plus utilisés, ont commencé à montrer leur âge. Ils sont tous deux composés d'une quantité significative de codes sources et de structures complexes qui les rendent difficiles à modifier et à mettre à niveau. La vitesse avec laquelle la nouvelle technologie se développe dans le domaine de la mécanique du bâtiment dépasse très souvent la capacité des groupes de soutien et de développement de la plupart des logiciels de simulation. Ainsi, le but recherché par les développeurs d'EnergyPlus était de développer un logiciel de simulation ralliant les forces et les capacités de calculs de Blast et DOE-2E et d'utiliser une structure de codes conviviale.

Les codes sources du logiciel EnergyPlus ont été séparés en de multiples modules. La modularité des codes sources du logiciel permet désormais aux développeurs et aux différents groupes de recherche de développer de nouveaux modules sans interférer aux autres modules en cours de développement. Cette nouvelle structure de codes sources favorise donc l'utilisation répandue et le large développement de modules auxiliaires. Aux dires des développeurs d'EnergyPlus, le succès de ce dernier repose sur l'intérêt soutenu et grandissant des groupes de recherche face au logiciel, de sorte que les modèles numériques développés par ceux-ci soient ajoutés aux codes sources du logiciel. Ainsi, la modularité et les améliorations de structure d'EnergyPlus devraient résoudre les problèmes de ressources intellectuelles et de développement continu des codes sources.

L'orientation axée sur le développement et la recherche scientifiques prise par les développeurs du logiciel EnergyPlus, m'a aidé à la formulation du sujet de mon mémoire de maîtrise autour de ce même logiciel. Deux orientations différentes ont été prises de ma part. La première portant sur l'analyse des capacités du nouvel outil de simulation (EnergyPlus) et la deuxième sur le développement et l'intégration d'un module de simulation aux modules existants. Pour commencer, il est essentiel d'identifier les recherches qui ont été faites. Étant donné que la première version du logiciel EnergyPlus est sortie en avril 2001, peu d'articles ont été écrits sur celui-ci. L'idée d'établir ma recherche en utilisant ce nouvel outil de calcul devenait donc des plus intéressantes. Ce chapitre présente dans un premier temps les validations des algorithmes de calculs effectuées sur le logiciel EnergyPlus et dans un deuxième temps les explications et justifications liées à la séparations en deux parties de mon mémoire de maîtrise.

## 1.1 Validation des algorithmes de calculs

Le département d'énergie du gouvernement américain a émis en décembre 2001, deux rapports d'étude comparative du logiciel EnergyPlus. Le premier rapport intitulé « EnergyPlus Testing with ANSI/ASHRAE Standard 140-2001 (BESTEST) »[1] présente les résultats d'une comparaison entre les logiciels EnergyPlus ver. 1.0.0.023, ESP-RV8, Blast3.0-193, Blast3.0-334, DOE-2D, DOE-2E, SERIRES/SUNCODE 5.7, SERIRES 1.2, S3PAS, TRNSYS 13.1 et TASE. Le test comparatif est effectué à partir du standard 140-2001, ANSI/ASHRAE qui traite uniquement du transfert de chaleur à travers l'enveloppe du bâtiment. Il est à noter que ce test ne permet pas de définir la précision absolue des résultats de calculs effectués par un logiciel de simulation, mais permet de comparer un nouveau logiciel, tel EnergyPlus, avec ses prédecesseurs, qui pour leur part, ont déjà fait leurs preuves. Cette première étude a permis de valider avec succès les algorithmes de calculs d'EnergyPlus.

Le deuxième rapport intitulé « EnergyPlus Testing with HVAC BESTEST »[2] présente les résultats de comparaison entre EnergyPlus ver. 1.0.0.023 et trois modèles analytiques établis et reconnus par l'agence internationale d'énergie (IEA), puis les résultats de comparaison entre EnergyPlus ver. 1.0.0.023, CA-SIS v.1, CLIM2000 2.1.6, DOE-2.1E-088, DOE-2.1E-133, PROMETHEUS et TRNSYS 14.2-TUD. Le test « HVAC BESTEST » permet d'évaluer la performance des algorithmes de calculs pour une zone climatisée à l'aide d'une unité de climatisation munie d'un serpentin de refroidissement à expansion directe. La comparaison entre les modèles analytiques et le logiciel EnergyPlus a permis d'éprouver ce dernier et d'identifier une erreur de l'ordre de 3% au niveau du calcul du rapport de la chaleur sensible sur la chaleur totale à l'intérieur de la zone climatisée. Comme dans le cas de la première étude, la comparaison du logiciel EnergyPlus avec ses prédecesseurs a permis de valider avec succès les algorithmes de calculs d'un système CVCA et de l'enveloppe.

Les deux rapports d'analyse du département d'énergie américain ont donc permis de valider les algorithmes de calculs utilisés par EnergyPlus en comparant les résultats de simulation obtenus par celui-ci avec des modèles analytiques et des résultats de simulations effectuées à l'aide d'autres logiciels de simulation reconnus et approuvés par l'agence internationale d'énergie (IEA). Ma recherche permet donc d'enrichir les résultats de ces deux rapports en validant les résultats de simulations EnergyPlus avec une base de données expérimentales.

## 1.2 Deux parties distinctes

Comme mentionné précédemment, mon mémoire de maîtrise se divise en deux parties distinctes. La première partie porte sur l'analyse des capacités de modélisation d'un système de traitement de l'air et des performances du logiciel EnergyPlus. On présente dans la seconde partie, les étapes d'élaboration et d'intégration d'un module de simulation dans le logiciel EnergyPlus. Dans le but de définir de façon précise et logique

le déroulement et le cheminement fait, on retrouve une méthodologie propre à chaque partie. Il est à noter également que les versions du logiciel EnergyPlus utilisées sont différentes dans les deux parties. Dans la première partie, la version 1.0.0.023, sortie en juin 2001, a été utilisée alors que dans la deuxième partie la version 1.0.2.008, sortie en juillet 2002, a été privilégiée.

## **CHAPITRE 2**

### **ANALYSE DES CAPACITÉS DU LOGICIEL ENERGYPLUS**

Ce chapitre présente l'analyse des capacités et des performances de calculs du logiciel EnergyPlus. Pour ce faire, deux comparaisons y sont présentées. La première comparaison a permis de valider les résultats obtenus par le logiciel EnergyPlus version 1.0.0.023 avec les données enregistrées sur le bâtiment simulé. La seconde comparaison a consisté à confronter les résultats obtenus par le logiciel EnergyPlus version 1.0.0.023 aux résultats du logiciel DOE-2E. Les détails sur les données introduites, les résultats obtenus et l'expérience de la modélisation à partir d'EnergyPlus sont présentés dans ce chapitre. Pour finir, certaines nouvelles particularités et capacités du logiciel par rapport à DOE-2 ont pu être testées, tel que le transfert de l'air du corridor adjoint à une salle de toilettes où l'on retrouve une évacuation d'air vicié. L'ensemble des commentaires sur la programmation et les résultats obtenus de ces deux derniers ajouts y sont également présentés.

#### **2.1 Objectifs**

Les principaux objectifs de cette première partie de mon mémoire sont :

- a. Valider un modèle développé à l'aide d'EnergyPlus en comparant les résultats de simulation avec une base de données préenregistrées sur le système simulé ;
- b. Comparer les capacités d'EnergyPlus avec celles d'un logiciel de référence ;
- c. Identifier les caractéristiques et particularités du logiciel rencontré lors de la modélisation.

## 2.2 Méthodologie

La méthodologie utilisée pour atteindre les objectifs mentionnés ci-dessus peut être résumée de la façon suivante :

- a. Choisir un système CVCA dont le fonctionnement et les caractéristiques mécaniques sont connus et précis ;
- b. Enregistrer les paramètres d'états et de fonctionnement (température, débit, etc.) de certaines zones sélectionnées et du système CVCA pour une période d'un mois ;
- c. Modéliser le plus précisément possible à l'aide d'EnergyPlus le système sélectionné et les zones qu'il alimente ;
- d. Valider les résultats de la simulation avec les données enregistrées ;
- e. Modéliser le système et les zones qu'il dessert à l'aide du logiciel DOE-2E ;
- f. Ajuster les deux modèles de simulation de façon à simuler l'opération du système de façon identique ;
- g. Comparer les résultats des simulations effectuées à l'aide des deux logiciels ;
- h. Décrire les éléments positifs et les lacunes du logiciel EnergyPlus rencontrés lors de la modélisation et de l'analyse des résultats.

## 2.3 Modélisation

Le bâtiment étudié est situé au centre-ville de Montréal et abrite l'École de Technologie Supérieure (ÉTS). On y retrouve deux sous-sols servant de stationnement, un rez-de-chaussée et 3 étages. L'édifice est équipé de 11 systèmes de traitement de l'air (CVCA) à volume d'air et température d'alimentation variables. Ils sont alimentés en eau chaude par le biais de la centrale thermique de la corporation de chauffage urbain de Montréal (CCUM) et en eau froide par le biais d'un groupe refroidisseur centrifuge de 800 tonnes. Des corps de chauffes électriques assurent, quant à eux, le chauffage des zones périphériques. Ce bâtiment, étant équipé de la toute dernière technologie en matière de

contrôle, a été choisi pour ses capacités d'enregistrement des données de fonctionnement de ses systèmes CVCA.

Pour limiter la quantité de données à saisir, 1 système sur 11 a été programmé et simulé, soit le système UTA-6. Il alimente 54 zones internes du 2<sup>ième</sup> étage de l'édifice. Ce système a été privilégié aux autres pour les raisons suivantes :

- a. Le fichier météorologique complet (ensoleillement, vitesse et orientation des vents, etc.) n'était alors pas disponible, d'où le choix d'un système alimentant des zones internes seulement ;
- b. Les zones alimentées sont principalement composées de salles de cours, de laboratoires informatiques et de laboratoires pratiques, dont l'horaire d'occupation est connu.

Tel que mentionné précédemment, le système UTA-6 alimente 54 zones thermiques. Malheureusement, le logiciel EnergyPlus limite à 50 le nombre de zones connectées à un même système de traitement. En fait, le nombre maximum de zones est limité par les éléments « Zone Mixer » et « Zone Splitter ». L'élément « Zone Splitter » permet de diviser le débit d'air en provenance du système et distribuer à chaque zone le débit d'air requis par celles-ci. L'élément « Zone Mixer » permet, quant à lui, de capter le débit d'air de retour des zones connectées à l'élément et d'acheminer l'air au retour du système. Étant donné que les zones thermiques de type corridor ont tous le même profil de charges internes, celles-ci ont donc été regroupées en une seule zone thermique. Cette mesure a permis d'abaisser le nombre de zone à 49 et ainsi contrer les limites du logiciel. Il est à noter que le zonage utilisé, lors de la saisie des données pour fin de simulation, a été calqué sur le zonage existant du bâtiment. Une zone est définie ici comme une pièce ou un groupe de pièces desservies par une même boîte à volume d'air variable (VAV). De plus, les débits maximums et minimums d'alimentation de chaque zone ont été fixés à partir du rapport d'état des boîtes à volume d'air variable fourni par le service des équipements de l'ÉTS.

Le schéma type d'un système de traitement de l'air VAV est présenté dans la figure 1. Dans la suivante, c'est-à-dire la figure 2, on retrouve le schéma de l'installation du système secondaire UTA-6 et du système primaire incluant le nom des différents nœuds modélisés dans EnergyPlus. Sur la figure, on remarque la présence de serpentins de réchauffage terminaux à la sortie des boîtes à volume d'air variable (VAV). Lorsqu'on définit dans le logiciel EnergyPlus une boîte à volume d'air variable, on doit automatiquement définir un serpentin de réchauffage. Étant donné que les boîtes VAV du système UTA-6 ne sont pas équipées de serpentins de réchauffage, la puissance du serpentin de réchauffage de chaque zone a été fixée à 0kW.

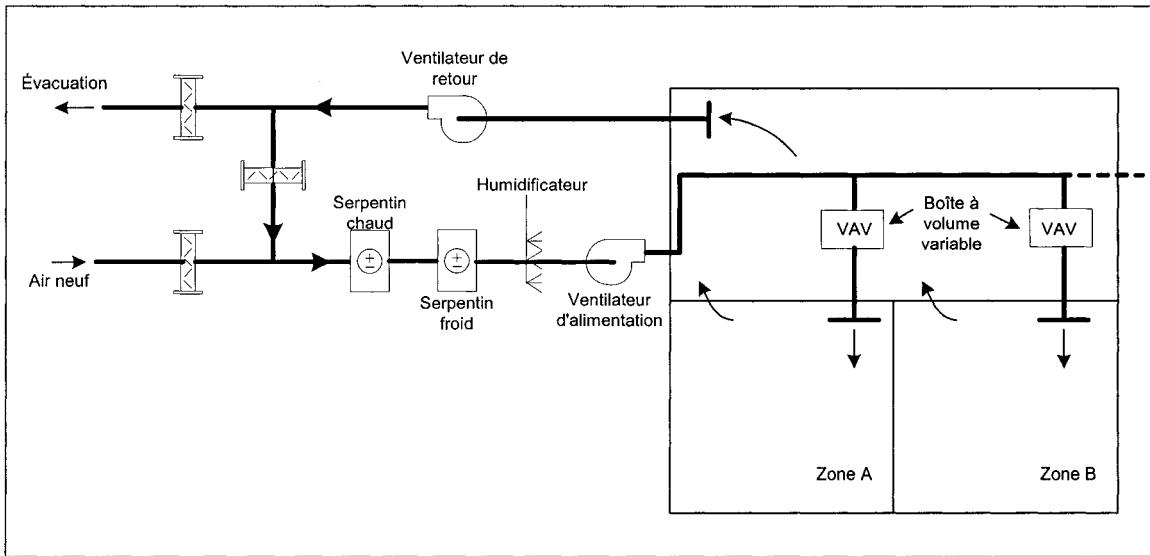


Figure 1 Schéma d'un système CVCA à volume d'air variable

L'ensemble des caractéristiques des composantes du système secondaire UTA-6 et du système primaire ont été introduites dans le modèle de simulation en appliquant les données fournies par les différents manufacturiers des équipements électromécaniques et les informations sur les plans et devis de construction. Les volets contrôlant l'admission d'air frais du système UTA-6 sont pilotés en fonction de l'enthalpie de mélange et du taux de CO<sub>2</sub> dans le conduit de retour. Étant donné que le contrôle de l'air frais par

l'entremise d'une sonde de CO<sub>2</sub> ne fait pas partie des modes contrôles disponibles dans EnergyPlus et que le traitement de l'air frais ne fait pas partie de l'analyse, l'admission d'air frais a donc été simplement fermée.

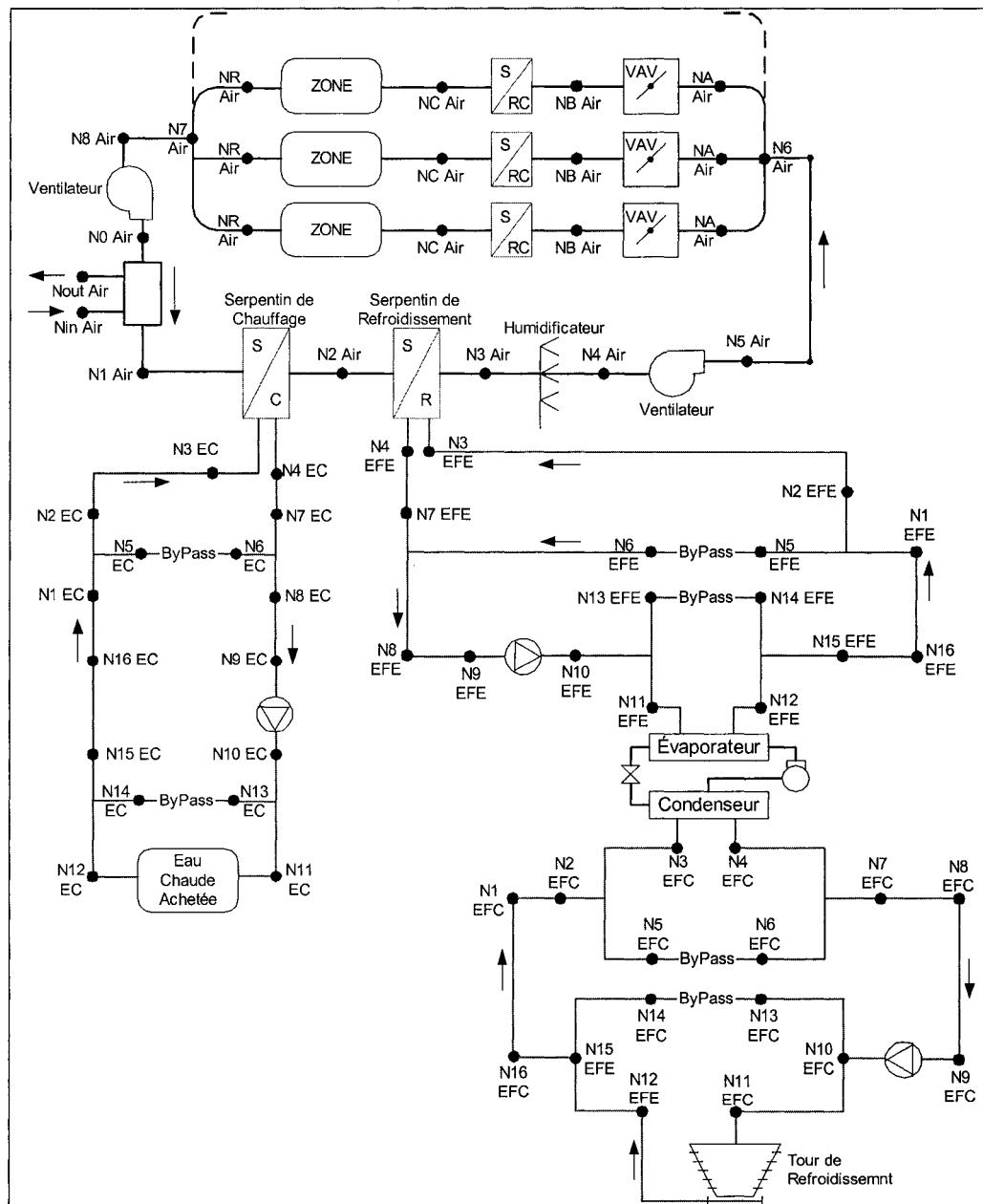


Figure 2 Schéma du système secondaire UTA-6 et du système primaire

Les horaires d'occupation des locaux ont été programmées en fonction de l'horaire des activités fournies par le décanat de la gestion des ressources de l'ÉTS. À partir de cet horaire, il était possible de connaître le nombre d'étudiants inscrits, l'horaire des activités et le type d'activités (cours, travaux pratiques ou laboratoires), et ce, pour chaque cours dispensé par les différents départements de l'ÉTS. Pour les locaux autres que les locaux à vocation académique, un relevé a été fait sur place afin de connaître le nombre d'occupants, le type d'activités et l'horaire d'occupation. Les horaires d'opération de l'éclairage ont été déterminées à l'aide des horaires d'occupation des locaux et des cédules de fermeture de l'éclairage de la commande centralisée du bâtiment. La puissance et le type d'éclairage des zones ont, quant à eux, été évalués en fonction des plans électriques de l'édifice. Les puissances d'équipements ont été fixées de façon sommaire. Par exemple, même si toutes les salles de cours sont équipées d'un projecteur vidéo, d'un projecteur d'acétates et d'un ordinateur, seul l'ordinateur a été considéré. Le projecteur vidéo et le projecteur d'acétates n'ont pas été considérés, car ils sont utilisés de façon ponctuelle. Pour les autres locaux, un relevé partiel d'équipements a été fait. À l'aide des informations données par les occupants, les horaires d'opération des équipements relevés ont pu être établis.

## 2.4 Validation du modèle

La validation, présentée ci-dessous, consiste en la comparaison des résultats obtenus par le logiciel EnergyPlus avec une base de données préenregistrées. La comparaison a été réalisée à deux niveaux (locaux desservis et système VAV). Les paramètres comparés au niveau des locaux desservis par le système VAV sont :

- a. La température ambiante ;
- b. Le débit d'air soufflé ;
- c. La température de soufflage.

Au niveau du système, seul le débit du ventilateur a été utilisé pour les fins de la comparaison. Le but de cette validation est d'obtenir la variation semblable des paramètres mentionnés précédemment obtenus par la simulation et l'enregistrement. La consommation d'énergie mensuelle ou annuelle, habituellement utilisée dans les validations de modèles énergétiques, n'a pas été prise en considération. En réalité, les données enregistrées n'ont pas permis d'isoler ces valeurs. De plus, les conditions climatiques extérieures complètes n'étaient pas disponibles et celles-ci, par le biais de l'admission d'air neuf au système, influencent la demande et la consommation d'énergie du système. Même si la base de données d'enregistrement contient beaucoup plus de paramètres, pour les fins de la validation du modèle, seuls les paramètres suivants ont été pris en considération :

- Au niveau des locaux présélectionnés :
  - ✓ La température ambiante ;
  - ✓ le débit d'air soufflé ;
  - ✓ la détection de l'occupation ;
  - ✓ la température extérieure.
- Au niveau du système :
  - ✓ La température d'alimentation ;
  - ✓ le débit du ventilateur ;
  - ✓ la demande de délestage ;
  - ✓ la température extérieure.

Certaines contraintes résultant du fonctionnement réel du système CVCA et des limitations propres au logiciel EnergyPlus ont engendré quelques ajustements sur le modèle de simulation et hypothèses de calculs qui sont présentés ci-dessous. Le pas d'enregistrement de la base de données est de 1 minute. Pour effectuer la comparaison, le pas de temps de calcul d'EnergyPlus a été fixé à son minimum, soit 10 minutes. La

température d'alimentation du système UTA-6 est contrôlée en fonction de la température extérieure et du débit du ventilateur. De plus, les gestionnaires du bâtiment, pour les fins de délestage, modifient les cédules de démarrage des systèmes primaires et secondaires du bâtiment ainsi que les différents points de consignes et séquences de contrôle (ex. : réduction de la vitesse de rotation des ventilateurs, arrêt du refroidisseur d'eau glacée, etc.).

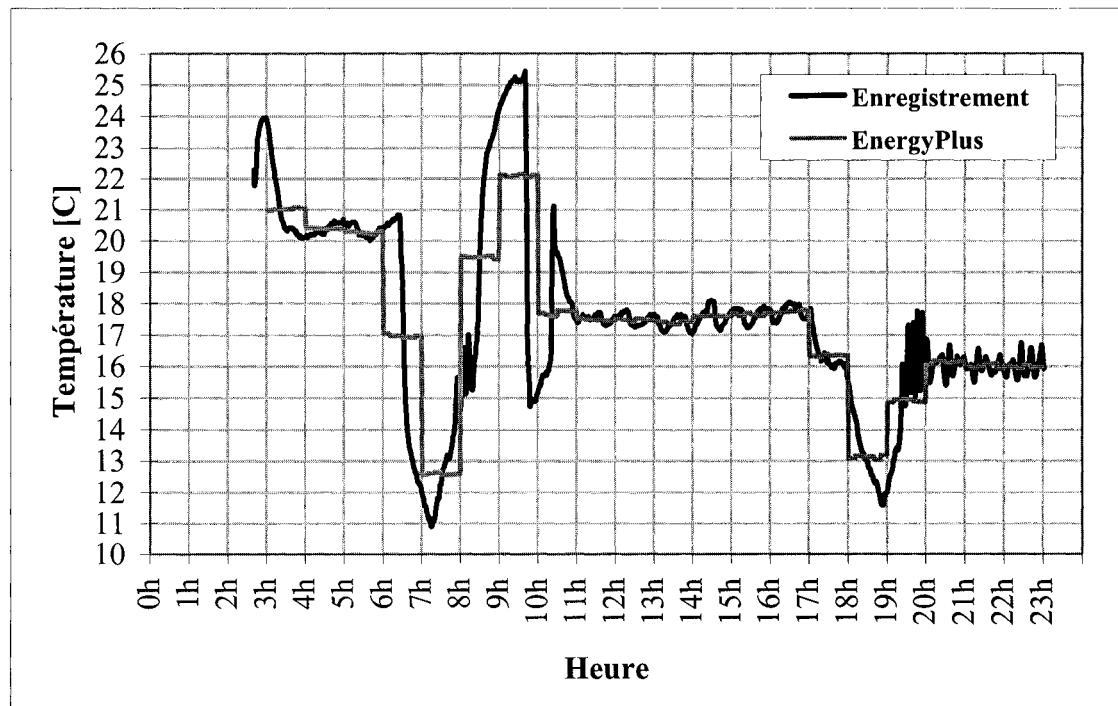


Figure 3      Variation de la température d'alimentation du système UTA-6 le 18 juin 2001

Par exemple, on remarque sur la figure 3 que la température d'alimentation du système UTA-6 fluctue beaucoup. Celle-ci atteint 11.0°C à 7h19 et 25.4°C à 9h40. Étant donné que la variation de température est influencée par divers facteurs, une cédule la gérant a été introduite dans le logiciel EnergyPlus. Malheureusement, le pas de temps des cédules pouvant être définies dans EnergyPlus est de 1 heure. La cédule a donc été générée en calculant la température moyenne pour chaque heure d'enregistrement (voir figure 3).

On remarque également que le système démarre à 2h39 et s'arrête à 23h, alors que la cédule introduite dans EnergyPlus fait démarrer le système à 3h et l'arrête à 23h.

Les figures 4 et 5 présentent respectivement la variation de la température ambiante et du débit volumétrique d'air soufflé dans le local 2304 obtenus par EnergyPlus et par le système d'acquisition de données. On peut constater une bonne concordance des courbes présentant les résultats de simulation et d'enregistrement. En effet, la différence moyenne absolue pour la température de pièce est seulement de  $0.36^{\circ}\text{C}$  (1.6%) et pour le débit de  $28.49 \text{ l/s}$  (12.7%). Il est à noter que les ajustements furent mineurs pour arriver à ces résultats grâce à un degré de véracité élevé des données relatives aux locaux étudiés. La différence de 12.7% du débit d'air soufflé est due en partie à la cédule de la température de soufflage utilisée et admise comme la moyenne horaire (mentionnée ci-dessus). La cédule de température de soufflage a pour effet d'augmenter ou de diminuer de façon tardive ou hâtive, par rapport aux données enregistrées, le débit de soufflage de la pièce ainsi que la température ambiante du local.

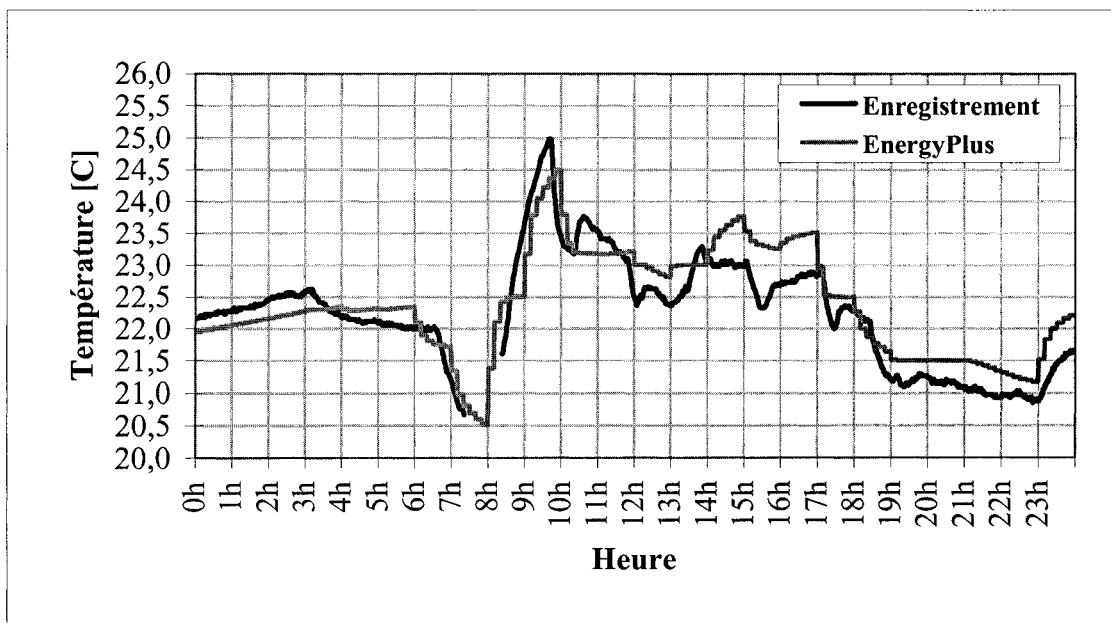


Figure 4 Variation de la température de la salle de cours 2304 le 18 juin 2001

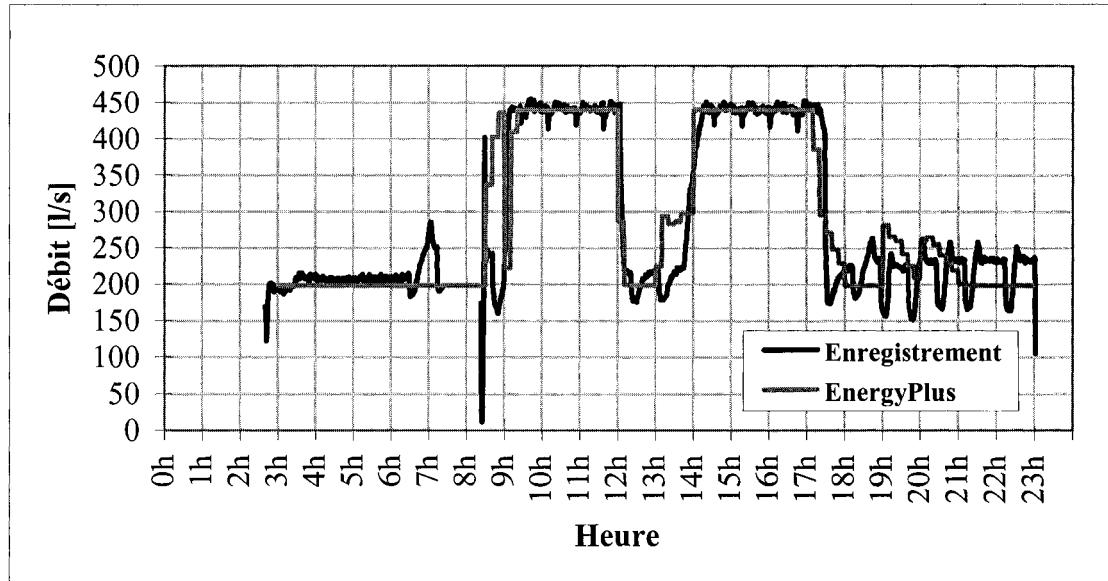


Figure 5 Variation du débit volumétrique d'air soufflé de la salle de cours 2304 le 18 juin 2001

Le choix de l'algorithme de calcul du coefficient de convection interne a une influence non-négligeable sur les conditions internes du local. Le modèle « *CeilingDiffuser* » est l'algorithme de calcul du coefficient de convection interne prévileged lors de la simulation. Ce modèle est basé sur les corrélations de convection forcée et mixte. Étant donné que le système de distribution de l'air est arrêté la nuit, il aurait été préférable d'utiliser le modèle « *Detailed* » qui est basé sur une corrélation par convection naturelle (voir les détails sur ce choix à la section 2.5).

L'analyse des résultats de simulation de l'opération du système de ventilation ne fut pas si concluante que celle au niveau des locaux desservis. En effet, la différence absolue observée pour la journée du 18 juin 2001, entre les résultats de simulation et la base d'enregistrement, est de l'ordre de 23% (voir la figure 6).

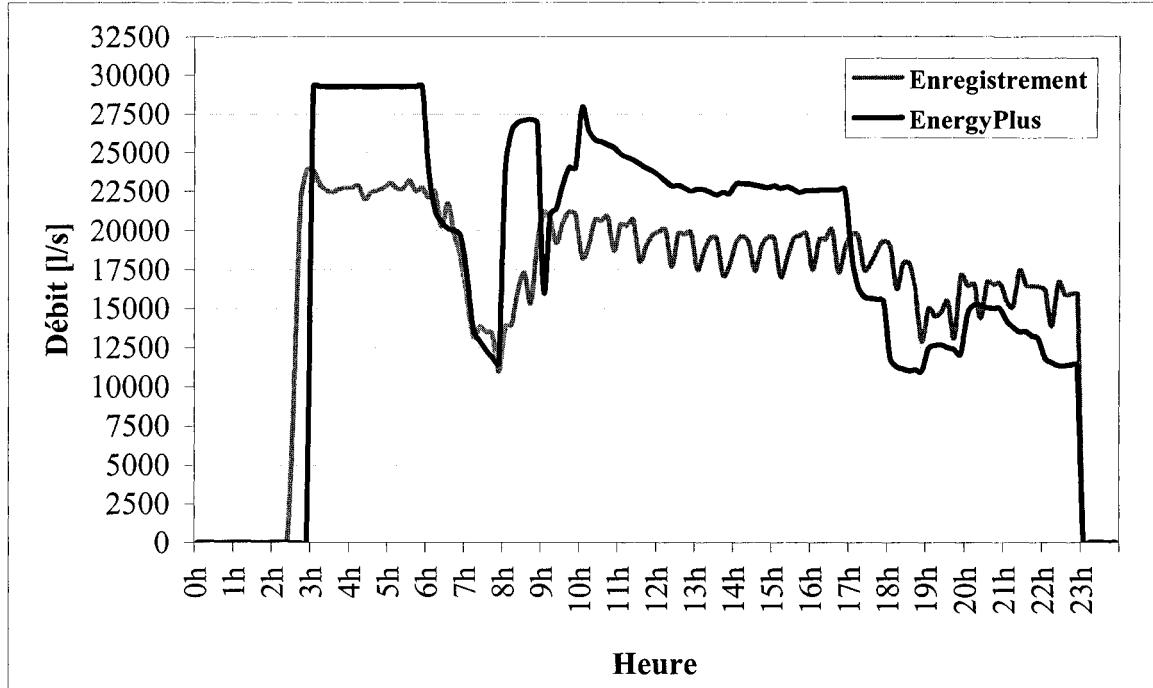


Figure 6 Variation du débit volumétrique d'air du ventilateur de l'unité UTA-6 le 18 juin 2001

Le faible degré de véracité des informations recueillies pour les locaux aux vocations autres qu'académique est la principale cause de la différence observée. Cependant, la validation de 4 zones du bâtiment avec succès permet de croire qu'avec un relevé détaillé accompagné d'un indice de véracité élevé de l'ensemble des zones desservies par le système, il serait possible d'obtenir les résultats escomptés. De plus, l'analyse comparative entre les résultats obtenues à l'aide d'EnergyPlus et ceux du logiciel DOE-2, reconnus pour l'exactitude de ces résultats, peut être considéré comme la validation des résultats de simulation d'EnergyPlus (voir la comparaison à la section 2.5).

## 2.5 Comparaison entre EnergyPlus et DOE-2

Afin de respecter les limites du logiciel DOE-2E et d'obtenir les mêmes conditions d'opération, le modèle de simulation fait sur EnergyPlus et validé dans l'étape précédente a été légèrement modifié et ajusté. À titre d'exemples :

- a. Afin de simplifier la simulation et l'analyse des résultats, la température d'alimentation du système a été fixée à 16°C et représente la moyenne enregistrée sur le système UTA-6 pour le mois de juin 2001 ;
- b. Le système de traitement de l'air a été programmé pour démarrer à 6h et pour arrêter à 23h ;
- c. Les fractions radiatives de la chaleur sensible dégagée par les occupants et les équipements ont dû être adaptées aux conditions prédéfinies par le logiciel DOE-2<sup>E</sup> ;
- d. Le choix de l'algorithme de calcul du coefficient de convection interne a également été modifié. Trois algorithmes de calculs du coefficient de convection interne sont disponibles dans la version 1.0.0.023 d'EnergyPlus. On y retrouve un algorithme simplifié (coefficient fixe), un détaillé (coefficient variable en fonction de l'inclinaison de la paroi et de la différence de température entre la paroi et le fluide en contact avec celle-ci) et un dernier utilisant un modèle de convection mixte et forcé pour la diffusion de l'air par le plafond.

La figure 7 présente la variation de la température ambiante en fonction du type d'algorithme utilisé. On remarque que le choix de l'algorithme a une influence non-négligeable sur les conditions internes du local 2304, et ce, surtout lorsque le local est inoccupé. Il est à souligner que l'algorithme simplifié (le seul disponible dans DOE-2E) a été choisi pour fin de comparaison, tandis que lors de la validation du modèle, l'algorithme utilisant un modèle de convection mixte et forcé s'est avéré le plus performant des trois algorithmes mentionnés ci-dessus.

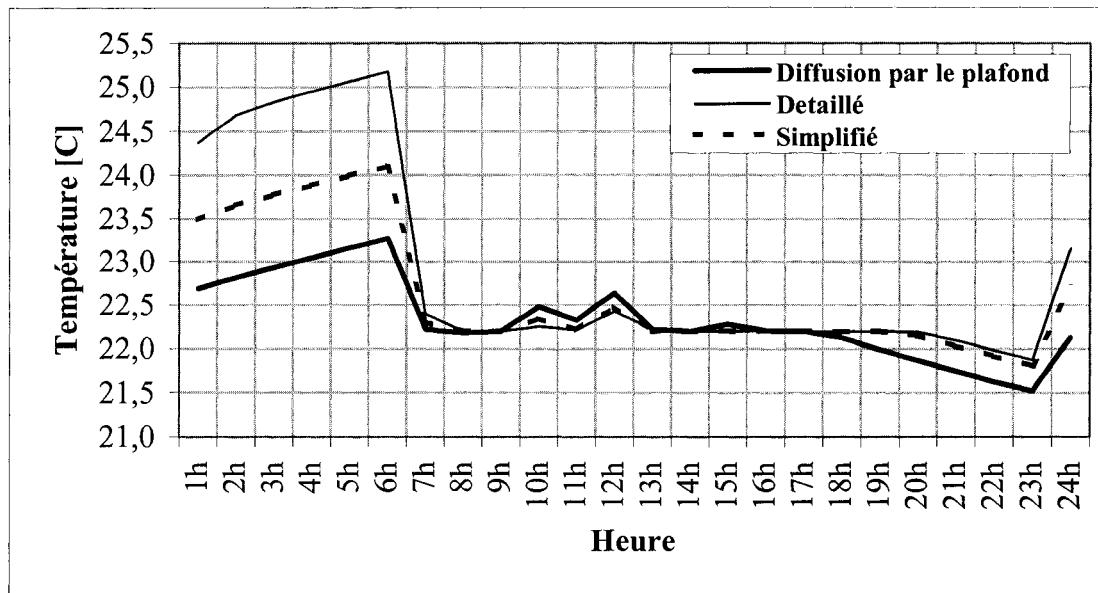


Figure 7 Influence du choix de l'algorithme de calcul du coefficient de convection interne sur la température ambiante du local 2304

Les figures 8 et 9 présentent les variations des températures et des débits volumétriques d'air soufflé dans le local 2304 obtenus à l'aide d'EnergyPlus et du logiciel DOE-2E. L'analyse des résultats démontre que la différence de température absolue du local entre les deux simulations est en moyenne pour une semaine de 0.24°C. De plus, on remarque que les deux profils journaliers du débit volumétrique sont semblables. La différence absolue moyenne observée pour une semaine entre les débits obtenus à l'aide d'EnergyPlus et de DOE-2E est de 9%. La principale différence observée porte sur le débit volumétrique d'air soufflé dans la zone lors du démarrage du système. En éliminant la première heure de fonctionnement du calcul de la moyenne, on remarque que l'erreur diminue de 2%.

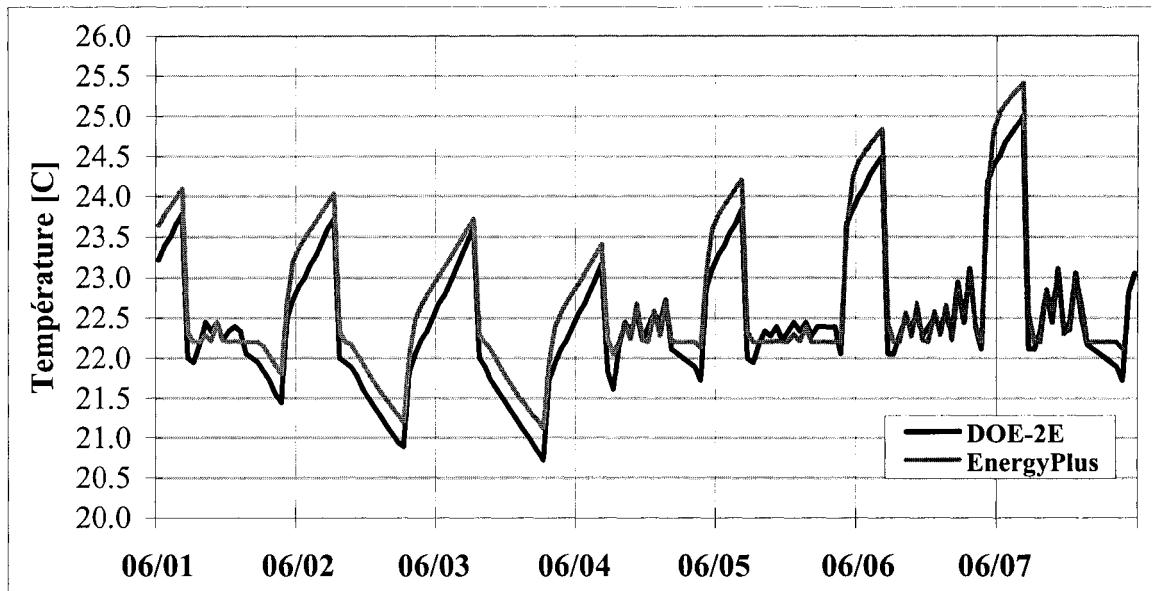


Figure 8 Répartition journalière de la température ambiante de la salle de cours 2304

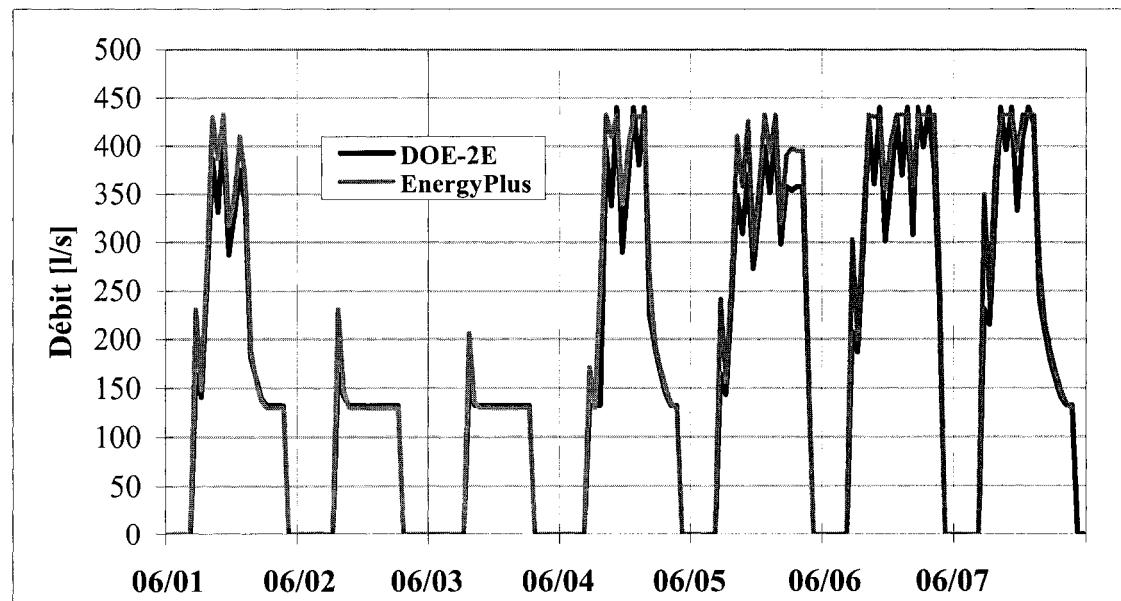


Figure 9 Répartition journalière du débit volumétrique d'air soufflé de la salle de cours 2304

Cette différence lors du démarrage est encore plus visible sur les figures 10 et 11 où l'on présente respectivement la variation du débit volumétrique total du ventilateur et la variation de la demande de refroidissement du système UTA-6, et ce, pour les journées du lundi et du mardi. Les différences absolues entre les débits du ventilateur et les demandes de refroidissement des simulations sont respectivement de l'ordre de 8.5% et 8.3%. Comme précédemment, si on soustrait la première heure de fonctionnement des systèmes, l'écart entre les débits est plutôt de 5.7% et de 7.9% entre les demandes. Cette différence est partiellement due au pas de temps de calculs de 10 minutes utilisé avec EnergyPlus alors que le pas de temps de calculs de DOE-2E est de 1 heure. Cependant, il est à noter que selon les données enregistrées sur le système UTA-6, le profil de la variation du débit du système ressemble davantage au profil obtenu à l'aide du logiciel EnergyPlus avec un pas de temps de calculs de 10 minutes.

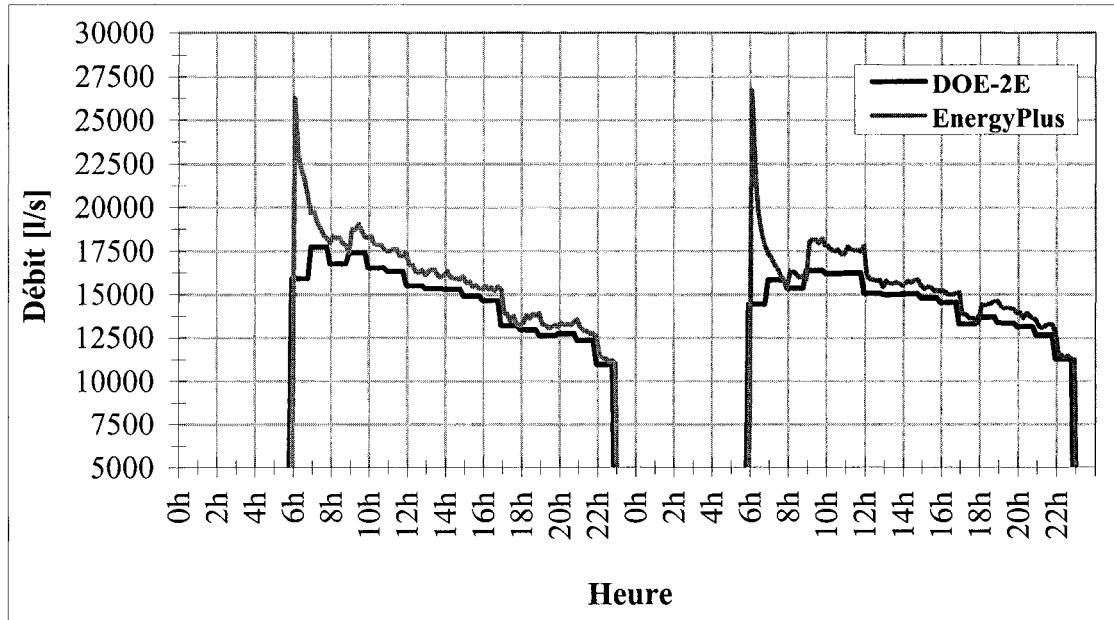


Figure 10 Répartition du débit volumétrique d'air du ventilateur du système UTA-6 pour les journées du lundi et mardi.

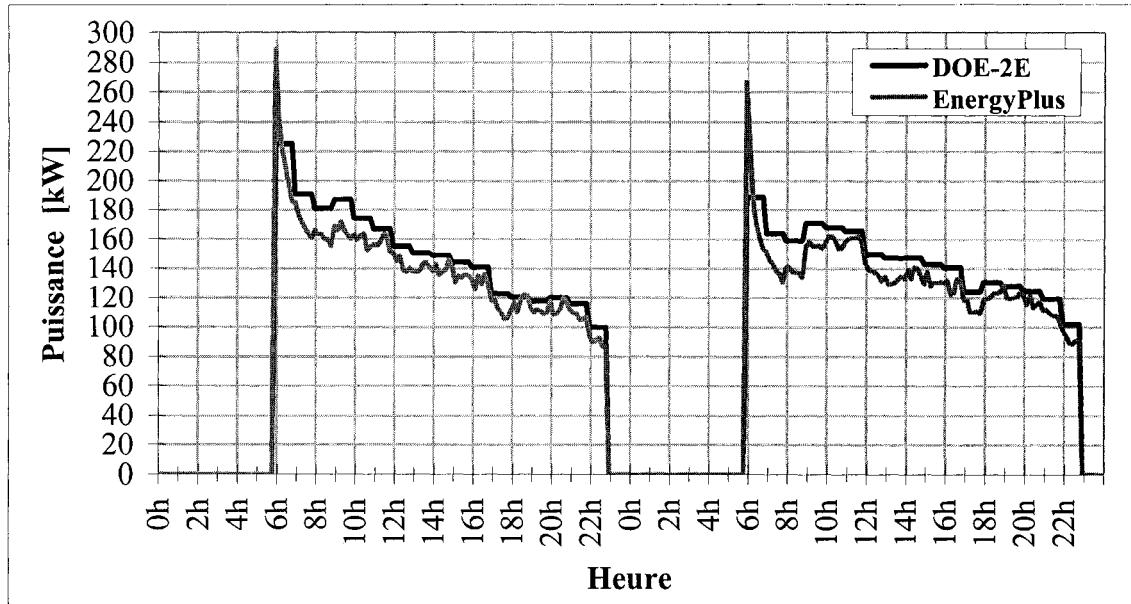


Figure 11 Répartition journalière de la demande de refroidissement au serpentin du système UTA-6 pour les journées du lundi et mardi.

## 2.6 Observations et anomalies

La présente section est un résumé des principales remarques et observations faites lors de la programmation du modèle dans la version 1.0.0.023 d'EnergyPlus et de l'analyse des résultats obtenus. Le but n'est donc pas de faire la liste complète des particularités du logiciel et de ces anomalies, mais de présenter celles rencontrées et observées.

### 2.6.1 Remarques et observations portant sur la modélisation

#### 2.6.1.1 Saisie de données

L'une des premières observations faites porte sur la quantité de données et de lignes codes à introduire dans EnergyPlus versus DOE-2E. Pour simuler le système UTA-6 et les zones qu'il alimente, environ 7500 lignes codes ont été introduites dans DOE-2 comparativement à 22000 lignes codes dans EnergyPlus. On trouve ci-dessous un

exemple typique de saisie de données dans DOE-2E versus EnergyPlus servant à définir l'occupation d'une zone thermique.

Exemple DOE-2E :

```
PEOPLE-SCHEDULE = Ced_occ_2102      $ Cédule d'occupation
NUMBER-OF-People = 31                  $ Nombre de personne
PEOPLE-HEAT-GAIN = 400                 $ Activité des occupants [W/pers.]
```

Exemple EnergyPlus :

```
PEOPLE,           !- Type d'objet
Local2102,        !- Nom de la zone
31,              !- Nombre de personne
Céd_Occ_2102,    !- Cédule d'occupation
0.70,            !- Fraction radiative de la chaleur sensible
Céd_Occ_Act,     !- Nom de la cédule d'activité [W/personne]
Local2102,        !- Nom du groupe de personnes
ZoneAveraged,    !- Type de calcul MRT (optionel)
,                !- Nom de la surface de référence (optionel)
Céd_Occ_Eff,     !- Cédule de l'efficacité du travail
Céd_Occ_Hab,     !- Cédule d'habillement (optionel) [clo]
Céd_Occ_Vit,     !- Cédule de vitesse de l'air (optionel) [m/s]
Pierce,          !- Type de rapport de confort thermique (optionel)
Ksu,             !- Type de rapport de confort thermique (optionel)
Anger;           !- Type de rapport de confort thermique (optionel)
```

Il est à noter que lors de la modélisation des zones desservies par le système UTA-6, l'ensemble des partitions internes ont été simulées dans les deux logiciels. Dans le logiciel EnergyPlus, une partition interne doit être définie dans les 2 zones contrairement à DOE-2E où l'on doit définir la partition seulement dans une des deux zones. Cette différence augmente ici considérablement le nombre de lignes codes dans EnergyPlus.

### 2.6.1.2 Définition des parois

La définition, dans EnergyPlus, d'une paroi (mur, toiture, plancher, fenêtre ou partition) impose l'introduction des coordonnées cartésiennes (x,y,z) de chaque extrémité de celle-ci comparativement à DOE-2E où il est possible d'introduire seulement la superficie, l'inclinaison et l'orientation de la paroi. Par contre, l'introduction des coordonnées

cartésiennes, grâce à un fichier *.dxf* créé par EnergyPlus, permet de visualiser les parois décrites dans celui-ci. La figure 12 présente l'ensemble des partitions modélisées dans EnergyPlus.

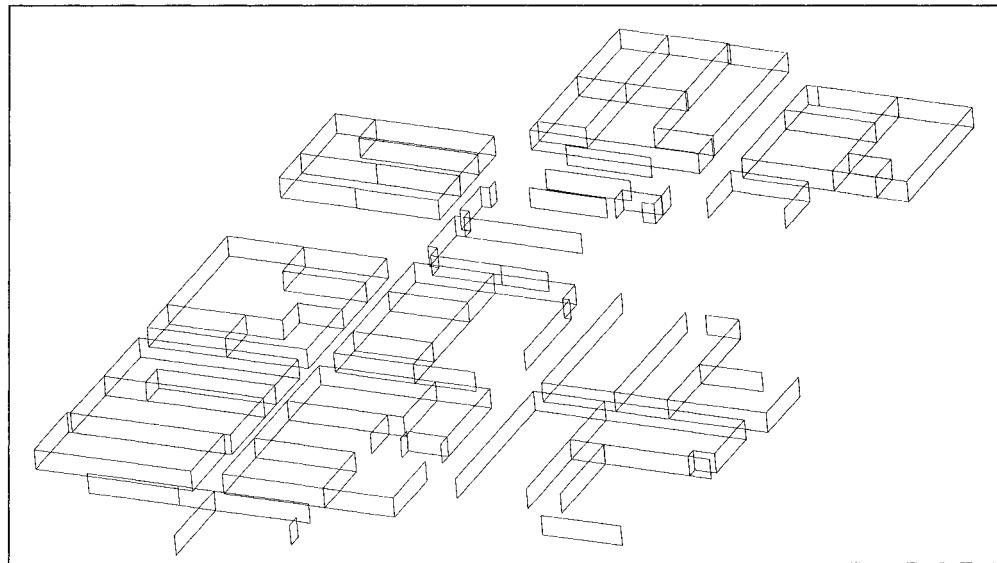


Figure 12      Partitions internes des zones desservies par le système UTA-6 et modélisées dans EnergyPlus

### 2.6.1.3 Outils de conversion

Les outils de conversion du code DOE-2 et des fichiers météorologiques ont été testés avec succès. Le premier outil permet de convertir la section LOAD du code DOE-2 incluant l'ensemble des horaires qui doivent être décrits de façon non-abrégée. Par la suite, l'utilisateur doit inclure les informations manquantes et requises par le logiciel EnergyPlus qui ne les sont pas avec DOE-2. Le second outil permet quant à lui de convertir les fichiers météos standard (ex. WYEC2) en fichier de type *.ewp*, utilisé par EnergyPlus.

#### 2.6.1.4 Agencement des systèmes mécaniques

Les différentes possibilités d'agencement des éléments d'un système rendent l'utilisation de ce logiciel intéressante lors de la simulation de systèmes dits non-conventionnels. Par exemple, modéliser deux systèmes de traitement de l'air desservant une même zone, soit une unité de climatisation de fenêtre avec un système central de traitement de l'air. La séquence de contrôle de ces unités est alors déterminée à partir des priorités en mode chauffage et refroidissement fixées par l'utilisateur. De plus, il est possible de modéliser le transfert de l'air d'une zone à une autre et ainsi de simuler les conditions ambiantes d'une zone climatisée de façon indirecte. Cette option a été testé sur les salles de toilettes qui étaient climatisées de façon indirecte par les corridors du 2<sup>ième</sup> étage (voir figure 13).

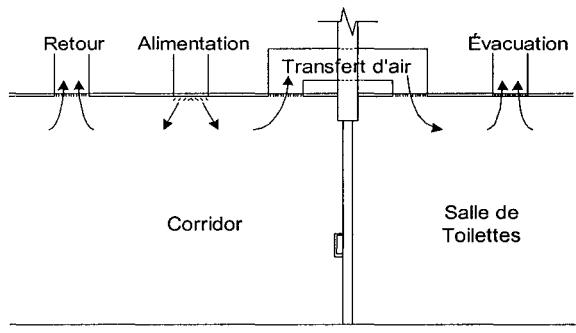


Figure 13      Climatisation indirecte des salles de toilettes

#### 2.6.1.5 Temps de calculs et précision des calculs

Les simulations ont été lancées sur un ordinateur muni d'un processeur AMD Athlon XP de 1000 Mhz et de 768 Meg. de mémoire vive. Pour la simulation d'un mois, avec un pas de temps de calcul de 10 minutes, le temps de calculs sur cet ordinateur était de 14 minutes sous EnergyPlus comparativement à quelques secondes avec DOE-2 (Pas de temps = 1h). La réduction du pas de temps de calculs permet d'évaluer de façon plus précise l'impact des variations importantes et subites de certains paramètres d'opération et/ou de fonctionnement des systèmes et des zones qu'il alimente. Cependant, les

cédules sont de type horaire ce qui réduit la précision des résultats obtenus. En observant les figures 14 et 15, on remarque l'influence du choix du pas de temps de calculs sur les demandes de refroidissement et de débit d'air du système de traitement de l'air UTA-6. Ainsi, on peut constater que la réduction du pas de temps, permet de mieux évaluer la dynamique du bâtiment étudié.

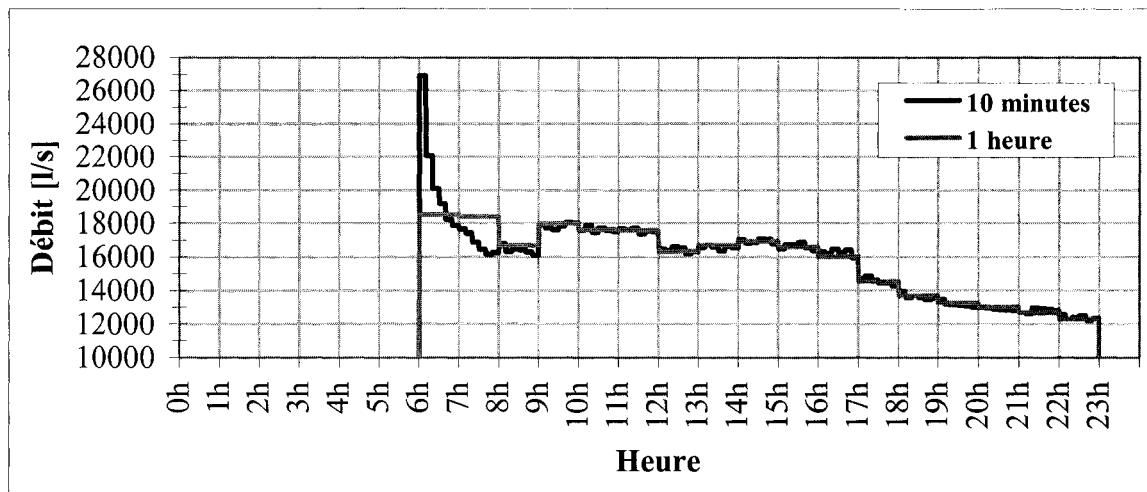


Figure 14 Variation du débit volumétrique d'air du système UTA-6 en fonction du pas de temps.

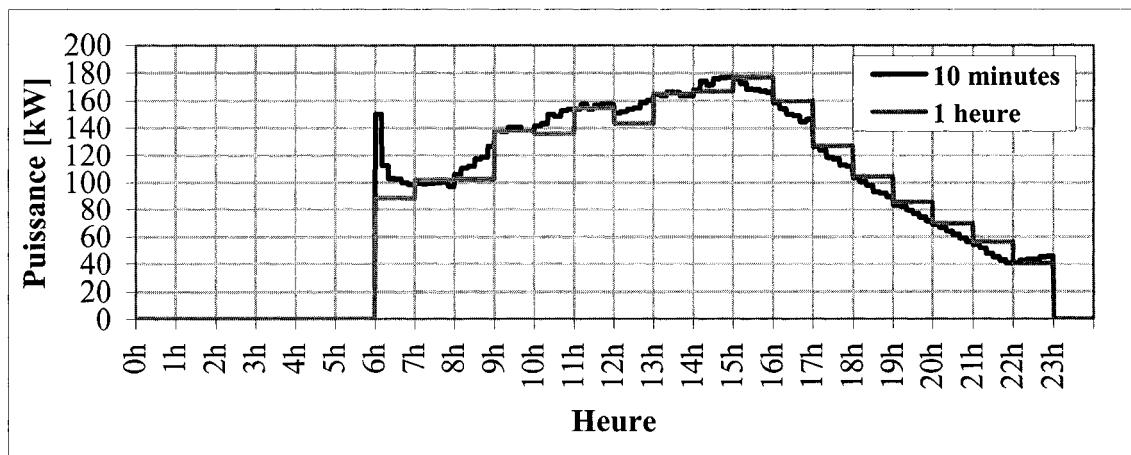


Figure 15 Variation de la demande de refroidissement du système UTA-6 en fonction du pas de temps.

### 2.6.1.6 Contrôle des éléments mécaniques

Le logiciel EnergyPlus a plusieurs limites et particularités du point de vue contrôle. Par exemple, le point de lecture du contrôleur de l'humidificateur ne peut être fait que par la lecture de l'humidité d'une zone et non d'un nœud du système de traitement de l'air (ex. : le retour d'air). De plus, l'utilisation d'un même nœud de lecture pour contrôler deux serpentins, soit un serpentin de chauffage et un serpentin de refroidissement, occasionne des erreurs dans la simulation (un des deux serpentins ne fonctionnera pas). L'ordre d'apparition des serpentins dicte l'ordre et l'emplacement des sondes de température (points de lecture). Par exemple, le point de lecture du contrôle de la température d'alimentation en mode refroidissement doit absolument être situé directement à la sortie du serpentin de refroidissement.

## 2.6.2 Anomalies rencontrées

### 2.6.2.1 Pertes de débit

Une perte de débit massique a été observée entre certains nœuds des systèmes CVCA. La provenance de ces pertes de débit n'a pu être décelée. Pour une journée type, la perte maximale observée a été de  $0.20181\text{kg/s}$ , soit  $0.83\%$  du débit total. Il est à noter que la quantité d'air perdue varie en fonction du temps.

### 2.6.2.2 Respect des limites

Lors de la simulation, le débit maximum du ventilateur d'alimentation du système VAV n'était pas respecté. Ainsi, lorsque la demande au ventilateur central augmentait et dépassait la capacité du ventilateur, le logiciel augmentait malgré tout le débit d'air pour répondre aux besoins de chacune des zones. Le débit volumétrique nominal du ventilateur d'alimentation était de  $23000\text{l/s}$  et la somme des débits des boîtes à volume

d'air variable était de 27060l/s. Ainsi, le facteur de diversité de 85% du système n'était pas respecté.

## CHAPITRE 3

### DÉVELOPPEMENT ET INTÉGRATION D'UN MODULE DE SIMULATION

Un des objectifs des concepteurs d'EnergyPlus était de développer le logiciel sur une plate-forme modulaire, permettant aux utilisateurs d'ajouter leur propre module de simulation aux modules existants. Pour se faire, les concepteurs ont utilisé le langage Fortran90. Un module est un paquet séparé de code source stocké dans un fichier séparé. Chaque module contient les algorithmes de calculs propres à un élément ou un groupe d'éléments étroitement liés. Par exemple, le module PlantPumps contient tous les codes sources permettant de simuler des pompes dans EnergyPlus. Ce module est contenu dans le dossier PlantPumps.f90. L'ensemble des modules, lorsqu'ils sont compilés, forment le fichier exécutable d'EnergyPlus. L'utilisateur désirant développer et intégrer un module de simulation au logiciel EnergyPlus, doit se procurer, auprès du département d'énergie américain, une licence de développeur et les codes sources d'EnergyPlus. EnergyPlus est composé de 126 modules Fortran. Pour arriver à intégrer un module de simulation aux autres modules, l'utilisateur doit comprendre l'ensemble des interactions entre chaque module. Un module de calculs permettant de simuler un mode de contrôle propre au Québec, soit la biénergie a été programmé et testé. Dans les années 80, Hydro-Québec a élaboré une nouvelle tarification encourageant les consommateurs d'énergie à utiliser deux sources de chaleur, soit l'électricité et un combustible (huile, gaz naturel, etc.). Cette nouvelle tarification avait pour but d'encourager la consommation d'électricité en période hors pointe ( $3.32¢/kWh$ ) et de pénaliser la consommation d'énergie en période de pointe ( $7.41¢/kWh$ ). Il est à noter que l'Hydro-Québec a abandonné le mode de tarification biénergie BT le 1<sup>er</sup> mai 1996 et que seuls les clients ayant un abonnement biénergie alors actif peuvent bénéficier de ce tarif. Depuis, la biénergie n'est disponible que pour les unités d'habitations (tarif DT). Le tarif DT s'applique à l'abonnement admissible au tarif D ou DM et en vertu duquel le client utilise, principalement à des fins d'habitation, un système biénergie. Un système biénergie est défini par Hydro-Québec

comme étant un système servant au chauffage de l'eau, de locaux ou à tout autre procédé de chauffe qui utilise l'électricité et un combustible comme sources d'énergie<sup>1</sup>.

### **3.1 Objectif**

L'objectif de la deuxième partie de mon mémoire est de développer, d'intégrer et de valider un nouveau module de simulation au logiciel de simulation EnergyPlus et d'évaluer les capacités d'adaptation du logiciel à l'intégration de nouveaux modules de simulation à ceux existants.

### **3.2 Méthodologie**

Les étapes de la méthodologie utilisée lors du développement du module de simulation ont été les suivantes :

- a. Identifier les règles de contrôle régissant la tarification biénergie selon Hydro-Québec.
- b. Identifier les contrôles existants et agissants sur systèmes mécaniques à contrôler.
- c. Étudier les modules existants afin de comprendre leurs relations avec les contrôles et les systèmes mécaniques.
- d. Programmer un module de simulation permettant d'agir et de contrôler les systèmes mécaniques selon le mode de contrôle biénergie.
- e. Intégrer au nouveau module de simulation, un outil diagnostique permettant d'alerter l'utilisateur lors de saisies non conformes.
- f. Valider le nouveau module de simulation à l'aide d'un modèle de simulation simple.
- g. Comparer et analyser des résultats de simulation.

---

<sup>1</sup> Règlement n°663 d'Hydro-Québec établissant les tarifs d'électricité et les conditions de leur application.  
Référence : Loi sur Hydro-Québec (L.R.Q.,chap.H-5)

### 3.3 Développement

#### 3.3.1 Règles de contrôle

Comme mentionné précédemment, pour qu'un client puisse bénéficier du tarif biénergie BT, il doit être en mesure d'utiliser l'électricité et un combustible comme sources de chauffage. De plus, le système biénergie doit satisfaire à toutes les conditions suivantes:

- a. Le système biénergie doit être muni des équipements de télécommande et de mesurage permettant d'appliquer la tarification en période de pointe et hors-pointe;
- b. Le système biénergie doit être conforme aux normes du distributeur (Hydro-Québec);
- c. La puissance maximale appelée au titre de l'abonnement ne doit en aucun cas excéder de plus de 10% la puissance installée des générateurs de chaleur électrique;
- d. La capacité du système biénergie en mode combustible doit être suffisante pour fournir toute la chaleur nécessaire au chauffage et aux procédés visés par le système biénergie.

Hydro-Québec entend par :

- a. Période de pointe : Toute période, le jour ou la nuit, lorsque la température extérieure est inférieure au seuil de température de transfert à l'exception de la période couverte par la plage horaire et toute période de reprise.
- b. Période hors pointe : Toute période, le jour ou la nuit, lorsque la température extérieure est supérieure au seuil de température de transfert, à l'exception de toute période de reprise
- c. Plage horaire : Une période de six heures et demie en 22h et 6h30.

La structure du tarif BT est la suivante :

Redevance mensuelle :

- 33.15 \$ plus

- 6.18 ¢ le kilowatt de puissance contractuelle. Le client souscrit par écrit une puissance contractuelle, qui ne peut être inférieure à 50 kilowatts. Cette puissance contractuelle correspond à au moins 85% de la puissance disponible, mais ne peut en aucun cas être supérieure à la puissance disponible. Si, au cours d'une période de consommation, la puissance maximale appelée excède de plus de 10% la puissance contractuelle, le distributeur applique à l'excédent une pénalité mensuelle de 12.78 ¢ le kilowatt.

Prix de l'énergie :

- 3.32 ¢ le kilowattheure pour toute l'énergie consommée pendant les périodes hors pointe, plus
- 7.41 ¢ le kilowattheure pour toute l'énergie consommée pendant les périodes de pointe, pendant les 25 premières heures d'utilisation, par période de 365 jours, de la puissance contractuelle pour l'option 1 ou pendant les 40 premières heures d'utilisation, par période de 365 jours, de la puissance contractuelle pour l'option 2. L'option 1 limite le nombre maximal d'heures d'application du prix en pointe pendant les périodes de pointe par période d'hiver à 400 heures par rapport à 600 heures pour l'option 2. De plus, la durée minimale d'applications du prix en pointe par jour et la durée minimale entre deux applications du prix en pointe pendant les périodes de pointe est de 2 heures pour l'option 1 et 4 heures pour l'option 2. Pour finir, le nombre maximal d'applications du prix en pointe par jour, pendant les périodes de pointes est fixé à 2 pour l'option 1 et à 1 pour l'option 2;
- 46.00 ¢ le kilowattheure pour toute l'énergie consommée en période de pointe.

### **3.3.2 Contrôles existants**

Par définition, les éléments contrôlés par un système biénergie sont : tout procédé de chauffage qui utilise l'électricité et un combustible comme sources d'énergie. La plupart des systèmes biénergie existants sont équipés d'une chaudière à eau chaude munies d'éléments électriques et d'un brûleur au gaz et/ou à l'huile. On retrouve différents

niveaux de contrôle des équipements primaires sur les boucles d'eau chaude dans EnergyPlus (voir la figure 16).

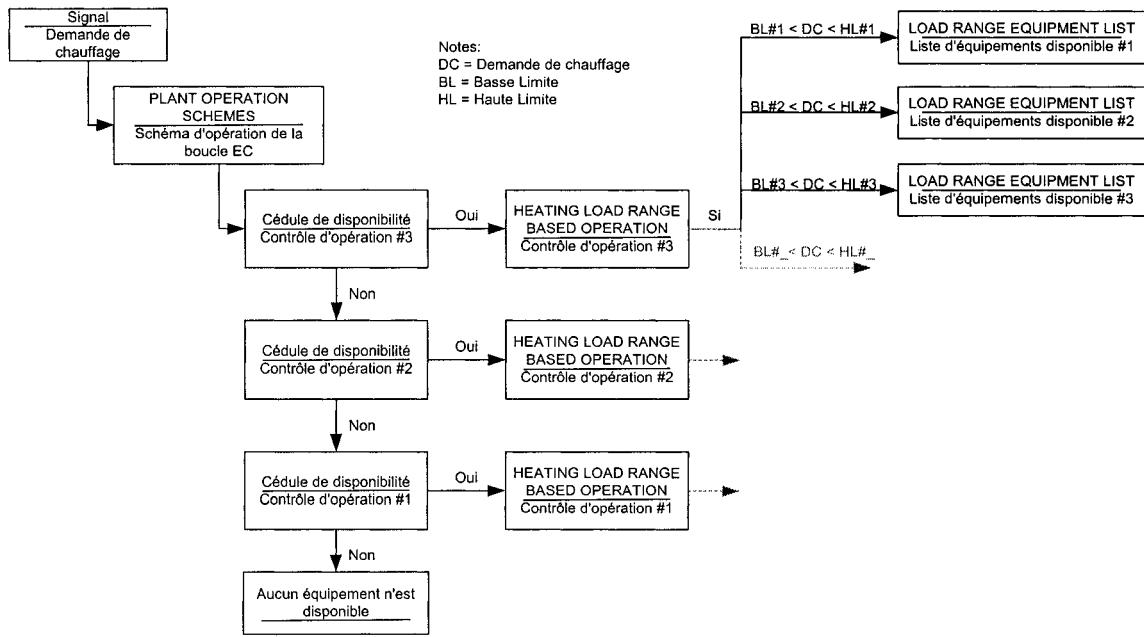


Figure 16 Plan de contrôle d'opération d'une boucle d'eau chaude dans EnergyPlus

Pour contrôler une boucle d'eau chaude, l'utilisateur doit établir le plan de contrôle d'opération de la boucle à partir de l'élément «*PLANT OPERATION SCHEMES*». Ce dernier est responsable de l'opération, des équipements raccordés et des priorités d'opération de la boucle. Le plan global d'opération comporte un nom d'objet, un nom d'identification qui est mis en référence dans le rapport principal de la boucle primaire qu'il contrôle et une liste de contrôles d'opération «*LOAD RANGE BASED OPERATION*». Chaque contrôle d'opération comporte un type de contrôle, un nom d'identification et une cédule de disponibilité. L'ordre d'apparition des contrôles dans la liste définit la priorité de chacun. Le dernier contrôle de la liste a priorité sur les autres, si ce dernier n'est pas disponible (en fonction de sa cédule de disponibilité), le contrôle précédent dans la liste est utilisé en fonction de sa cédule de disponibilité. En d'autres termes, si le dernier contrôle d'opération est disponible, il est employé par la simulation

pour définir comment la boucle fonctionne. S'il n'est pas disponible, le contrôle d'opération précédent ce dernier dans la liste est vérifié pour voir s'il est disponible. La recherche continue jusqu'à ce que l'arrangement prioritaire le plus élevé disponible soit trouvé.

Exemple :

PLANT OPERATION SCHEMES,	!- Nom de l'objet
Schéma d'opération de la boucle EC,	!- Nom d'identification
HEATING LOAD RANGE BASED OPERATION,	!- Type de contrôle d'opération #1
Opération fct demande chauffage#1,	!- Nom du contrôle d'opération #1
Dispo. du contrôle d'opération EC#1,	!- Nom de la cédule d'exploitation #1
HEATING LOAD RANGE BASED OPERATION,	!- Type de contrôle d'opération #2
Opération fct demande chauffage#2,	!- Nom du contrôle d'opération #2
Dispo. du contrôle d'opération EC#2,	!- Nom de la cédule d'exploitation #2
HEATING LOAD RANGE BASED OPERATION,	!- Type de contrôle d'opération #3
Opération fct demande chauffage#1,	!- Nom du contrôle d'opération #3
Dispo. du contrôle d'opération EC#1;	!- Nom de la cédule d'exploitation #3

Le contrôle d'opération de chauffage «*HEATING LOAD RANGE BASED OPERATION*» permet de définir les plages de charges d'opération et les équipements disponibles pour chaque plage de charges. Le contrôle d'opération comporte un nom d'objet, un nom d'identification (utilisé lors de la définition du plan global d'opération «*HEATING LOAD RANGE BASED OPERATION* ») et une liste de trios de données. Chaque trio est formé d'une limite inférieure et supérieure d'opération et du nom d'une liste d'équipements disponibles «*LOAD RANGE EQUIPEMENT LIST*».

Exemple :

HEATING LOAD RANGE BASED OPERATION,	!- Nom de l'objet
Opération fct demande chauffage#1,	!- Nom d'identification #1
0,	!- Basse limite #1 [W]
1000000,	!- haute limite #1 [W]
Liste d'équipements EC#1;	!- Nom de la liste d'équipements contrôlés

La liste d'équipements pour la plage de charge «*LOAD RANGE EQUIPMENT LIST*» est employée pour définir les équipements disponibles pour une plage de charge particulière et pour définir l'ordre d'utilisation de ceux-ci. La liste d'équipements comporte un nom d'objet, un nom d'identification (utilisé lors de la définition du contrôle d'opération «*LOAD RANGE BASED OPERATION*») et une liste de paires de données. Chaque paire est formée d'un type d'équipement primaire et d'un nom d'identification. La priorité d'utilisation de chaque équipement est dictée par l'ordre d'apparition des équipements primaires dans la liste.

Exemple :

LOAD RANGE EQUIPMENT LIST,	!- Nom de l'objet
Liste d'équipements EC#1,	!- Nom d'identification #1
BOILER:SIMPLE,	!- Type d'équipement #1
Chaudière électrique;	!- Nom de l'équipement #1

On remarque donc que les contrôles existants d'EnergyPlus permettent de contrôler les équipements primaires raccordés à une boucle d'eau chaude en fonction d'horaires de fonctionnement et de la demande de chauffage. Malheureusement, comme ses prédecesseurs, EnergyPlus ne permet pas de contrôler l'opération des équipements raccordés à une boucle d'eau chaude en fonction d'une température extérieure.

### **3.3.3 Modules existants**

Bien que les modules d'EnergyPlus aient été conçus pour être aussi indépendants que possible, évidemment ils ne peuvent l'être complètement. La structure globale du programme (niveau supérieur) est présentée ci-dessous.

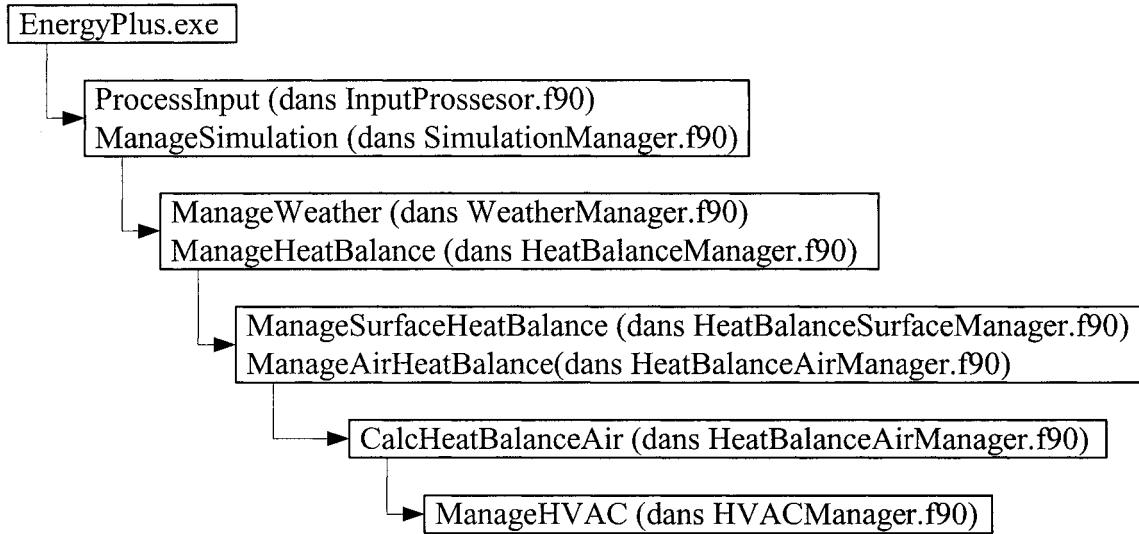


Figure 17 Structure interne du logiciel EnergyPlus – Niveau supérieur

Le module de simulation «*HVACManager*» est responsable d'appeler tous les modules liés au chauffage, à la ventilation et à la climatisation de l'air, et est divisé en plusieurs sous-routines de simulation. On retrouve des modules responsables d'appeler les modules liés aux systèmes de traitement de l'air (*SimAirServingZones.f90*), aux équipements de zone (*ZoneEquipment Manager.f90*), à l'énergie fournie et requise par les systèmes primaires (*PlantLoopSupplySideManager.f90* et *PlantDemandSideLoops.f90*) et à l'énergie fournie et requise du côté condenseur (*CondLoopManager.f90* et *CondenserDemandSideLoops.f90*). Il y a également des modules de simulation responsables de l'approvisionnement en électricité et en gaz. Pour chaque pas de temps de calcul, les routines de calculs contenues dans chaque module sont appelées à plusieurs reprises jusqu'à convergence de l'ensemble des calculs. L'arbre ci-dessous présente la structure partielle du module «*ManageHVAC*».

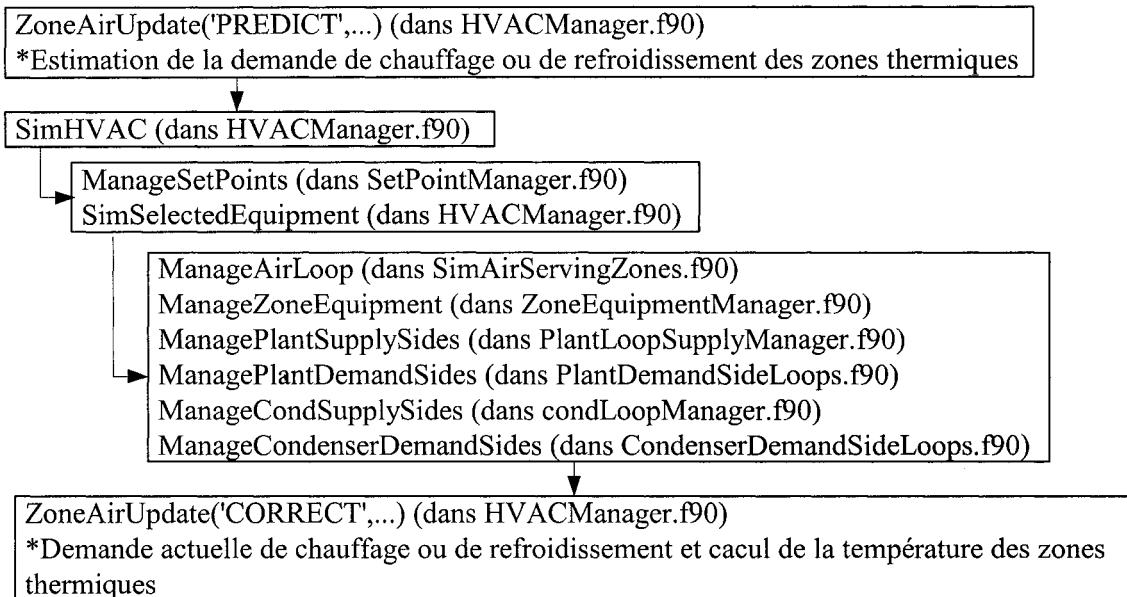


Figure 18 Structure partielle du module «*ManageHVAC.f90*»

L'ensemble des modules de simulations présentés ci-dessus a pour rôle de superviser l'appel des modules responsables du contrôle des différents éléments mécaniques des systèmes primaires et secondaires. Par exemple, le module «*PlantLoopSupplyManager.f90*» supervise l'appel du module «*PlantCondLoopOperation.f90*» qui contrôle les boucles d'eau chaude, d'eau froide et de rejet de chaleur. Ce dernier module devra être revu et adapté de façon à contrôler la boucle d'eau chaude et les systèmes raccordés en fonction des séquences de contrôle nécessaires à un système biénergie.

### 3.3.4 Nouveau module

Avant d'entreprendre le développement d'un module de contrôle, il faut identifier et définir les informations dont l'utilisateur du logiciel doit fournir lors de la programmation d'un modèle de simulation utilisant la biénergie. L'objet de saisie «*CONTROLE BIENERGIE*» a donc été ajouté aux objets existants. L'objet «*CONTROLE BIENERGIE*» comporte un nom d'objet, un nom d'identification, le nom

du plan global d'opération équipé du contrôle biénergie et la température limite d'opération, en degré Celcius, de l'équipement ayant comme source de chaleur l'électricité.

Exemple :

CONTROLE BIENERGIE,	!- Nom de l'objet
Contrôle Bi-Energie #1,	!- Nom d'identification
Schéma d'opération de la boucle EC,	!- Nom du plan global d'opération
-12;	!- Température limite [C]

L'objet «*CONTROLE BIENERGIE*» permettra à l'utilisateur d'identifier le plan global d'opération qui est équipé d'un contrôle bi-énergie et la température limite d'opération des systèmes de chauffage fonctionnant à l'électricité. L'utilisateur devra utiliser les horaires et l'ordre de priorité d'opération de l'objet «*LOAD RANGE BASED OPERATION*» pour élaborer la stratégie de contrôle biénergie. Afin de limiter les problèmes liés à l'intégration complète et totale d'un nouveau module de simulation, l'adaptation d'un module existant a été privilégiée. Ainsi, le module «*PlantCondLoopOperation.f90*» a été modifié de façon à répondre aux exigences de contrôle d'un système biénergie. L'étude complète de l'algorithme et des routines de calculs a été fait de façon à comprendre l'interaction du module «*PlantCondLoopOperation.f90*» sur les autres modules.

### **3.3.4.1 Modifications et ajouts**

Les modifications et les ajouts apportés au module «*PlantCondLoopOperation.f90*» sont expliqués et présentés ci-dessous.

Ajout #1 : Utilisation de la variable «*OutDryBulbTemp*» contenu dans module «*DataEnvironment.f90*» qui permet d'obtenir la température extérieure sèche pour chaque pas de temps de calcul.

```
! ****
USE DataEnvironment, ONLY: OutDryBulbTemp
! ****
```

Ajout #2 : Utilisation des fonctions «*GetNumObjectsFound*» et «*GetObjectItem*» contenues dans le module «*InputProcessor.f90*». La première fonction permet de compter le nombre de fois que l'on retrouve un type d'objet dans le fichier d'entrées et la deuxième fonction permet de connaître l'ensemble des renseignements d'un objet décrit dans le fichier d'entrées.

```
! ****
USE InputProcessor, ONLY: GetNumObjectsFound, GetObjectItem
! ****
```

Ajout #3 : Définition de l'ensemble des variables utilisées ultérieurement.

```
! ****
INTEGER :: OpNumBi      ! Compteur Bi-Energie
INTEGER :: NumOpBiEnergie ! Nombre total de contrôle biénergie
INTEGER :: NumAlphasBi
INTEGER :: NumNumsBi
INTEGER :: IOSTATBi
REAL, DIMENSION(30) :: NumArrayBi
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArrayBi
LOGICAL(1):: OpBiEnergieFound
INTEGER, SAVE :: ValeurBiEnergie
INTEGER :: TempBiEnergie
! ****
```

Ajout #4 : Calcul du nombre d'objet «*CONTROLE BIENERGIE*».

```
! ****
NumOpBiEnergie = GetNumObjectsFound('Controle BiEnergie')
! ****
```

Ajout #5 : Acquisition des renseignements d'entrées de l'objet «*PLANT OPERATION SCHEMES*».

```
! ****
CALL GetObjectItem('PLANT OPERATION SCHEMES',LoopNum,AlphArray,NumAlphas, &
                  NumArray,NumNums,IOSTAT)
! ****
```

Ajout #6 : Routine de calcul permettant d'associer l'objet «*CONTROLE BIENERGIE*» (si existant) à l'objet «*PLANT OPERATION SCHEMES*». Si l'association est possible, les paramètres de contrôle propre au contrôle biénergie sont appliqués au plan global d'opération de la boucle. Il est à noter, que le contrôle biénergie est appliqué seulement sur le dernier équipement dans la liste d'équipement de l'objet «*PLANT OPERATION SCHEMES*». Par exemple, pour un pas de temps de calcul donné, si la température extérieure est supérieure à la limite fixée par l'utilisateur, la dernière chaudière dans la liste d'équipements du plan global d'opération est mise en fonction conditionnellement à la cédule de disponibilité de cette dernière. Si la chaudière n'est pas disponible, la chaudière précédente dans la liste est mise en fonction conditionnellement à sa cédule de disponibilité. La recherche continue jusqu'à ce qu'une chaudière soit disponible. Par contre, si la température extérieure est inférieure à la température limite, la dernière chaudière dans la liste d'équipements du plan global d'opération est mise hors service et ce, peu importe la cédule de disponibilité de cette dernière, et la chaudière précédente dans la liste est mise en fonction conditionnellement à sa cédule de disponibilité. Évidemment, la recherche continue jusqu'à ce qu'une chaudière soit disponible.

```

! ****
PlantOpScheme = AlphArray(1)
OpBiEnergieFound = .FALSE.
DO OpNumBi = 1, NumOpBiEnergie
    CALL etObjectItem('Controle BiEnergie',OpNumBi,AlphArrayBi,NumAlphasBi, &
        NumArrayBi, NumNumsBi, IOSTATBi)
    IF (PlantOpScheme /= AlphArrayBi(2)) CYCLE
        TempBiEnergie = NumArrayBi(1)
        IF (OutDryBulbTemp < TempBiEnergie .AND. &
            Num == PlantSupplySide(LoopNum)%NumOpSchemes) THEN
            ValeurBiEnergie = 0
        ELSE
            ValeurBiEnergie = 1
        END IF
        OpBiEnergieFound = .TRUE.
    END DO
    IF (.not. OpBiEnergieFound) THEN
        ValeurBiEnergie = 1
    END IF

    IF (GetCurrentScheduleValue(PlantSupplySide(LoopNum)%OpScheme(Num)%SchedPtr) &
        * ValeurBiEnergie <= 0.0) CYCLE
! ****

```

Il est à noter que l'intégralité du module de simulation est présenté en annexe. L'utilisateur désirant simuler la biénergie selon le mode de contrôle d'Hydro-Québec devra définir parmi les systèmes primaires une chaudière électrique et une chaudière à combustible (ex : gaz). Lors de l'élaboration du plan global d'opération de la boucle d'eau chaude, la chaudière électrique devra être la dernière dans la liste d'équipements, la chaudière à combustible devra la précéder et la chaudière électrique devra réapparaître dans la liste juste avant la chaudière à combustible. De plus, la chaudière électrique devra, selon sa cédule de disponibilité, être disponible 24 heures sur 24 et la cédule de disponibilité de la chaudière à combustible devra empêcher la chaudière de fonctionner de 22h à 6h.

### 3.3.4.2 Outils diagnostiques

Dû à la complexité de la structure de programmation, les utilisateurs du logiciel EnergyPlus peuvent à l'occasion utiliser des objets de façon non-conforme ou encore oublier des objets essentiels au bon fonctionnement de leur modèle de simulation. Pour cette raison, les développeurs d'EnergyPlus ont ajouté aux routines de calculs des outils diagnostiques qui permettent de cibler la provenance d'erreurs de programmation. Les utilisateurs du logiciel peuvent donc être guidés dans le débogage de leur modèle de simulation. Un outil de diagnostic a donc été ajouté au module de simulation intégrant le contrôle biénergie. Les lignes de codes ajoutées au module « PlantCondLoopOperation.f90 » sont expliqués et présentés ci-dessous.

Ajout #1 : Utilisation des fonctions «*ShowWarningError*» et «*ShowSevereError*» contenues dans le module «*DataGlobals.f90*». La fonction «*ShowWarningError*» permet d'enregistrer un message d'avertissement dans un fichier ASCII ayant comme descriptif le nom du fichier d'entrée et l'extension «*err*». L'utilisateur peut à la fin de la simulation visualiser l'ensemble des avertissements contenues dans le fichier. La fonction

«*ShowSevereError*» permet d'arrêter la simulation et d'enregistrer un message d'erreur dans le même fichier ASCII. Par la suite, l'utilisateur peut trouver la source de l'erreur dans la simulation en consultant le fichier ASCII.

```
! ****
USE DataGlobals, ONLY: ShowWarningError, ShowSevereError
! ****
```

Ajout #2 : Définition de l'ensemble des variables utilisées ultérieurement.

```
! ****
INTEGER :: OpNumBi      ! Compteur Bi-Energie
INTEGER :: NumOpBiEnergie ! Nombre total de controle BiEnergie
INTEGER :: NumAlphasBi
INTEGER :: NumNumsBi
INTEGER :: IOSTATBi
REAL, DIMENSION(30) :: NumArrayBi
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArrayBi
LOGICAL(1):: OpBiEnergieFound
INTEGER, SAVE :: ValeurBiEnergie
INTEGER :: TempBiEnergie
! ****
```

Ajout #3 :Calcul du nombre d'objet «*CONTROLE BIENERGIE*».

```
! ****
NumOpBiEnergie = GetNumObjectsFound('Controle BiEnergie')
! ****
```

Ajout #4 : Initialisation d'un test logique (intégration à l'intérieur d'une boucle de calcul existante – voir annexe)

```
! ****
OpBiEnergieFound=.FALSE.
! ****
```

Ajout #5 : Test logique qui permet de vérifier l'existence du plan global d'opération spécifié dans l'objet «*CONTROLE BIENERGIE*» dans le fichier d'entrées. Si le test échoue, la simulation est arrêtée et un message d'erreur est envoyé à l'utilisateur via un fichier ASCII. Par contre, si le plan global d'opération existe, la simulation continue et

un message, confirmant la présence du contrôle biénergie sur le plan global d'opération concerné, est envoyé à l'utilisateur via un fichier ASCII.

```

! ****
DO OpNumBi = 1, NumOpBiEnergie
CALL GetObjectItem('Controle BiEnergie',OpNumBi,AlphArrayBi,NumAlphasBi, &
NumArrayBi, NumNumsBi, IOSTATBi)
    IF (PlantOpScheme /= AlphArrayBi(2)) CYCLE
    OpBiEnergieFound=.TRUE.
END DO

IF (.Not. OpBiEnergieFound) THEN
IF (NumOpBiEnergie >= 1) THEN
IF (NumOpSchemes == OpNum) THEN
CALL ShowSevereError('Controle Bi-Energie sur://'TRIM(AlphArrayBi(2)))
ErrorsFound=.true.
END IF
    END IF
END IF

IF (OpBiEnergieFound) THEN
CALL ShowWarningError ('Controle Bi-Energie sur://'TRIM(PlantOpScheme))
END IF
! ****

```

Comme mentionné précédemment, l'ensemble du module de simulation est présenté en annexe. Tous les modules de simulation ont été compilés à l'aide du compilateur « Visual Fortran 6.5 ». Les options suivantes ont été sélectionnées lors de la compilation des modules dans « Visual Fortran 6.5 »:

- Génération de code
  - ✓ Génération de code optimisé
  - ✓ Librairie mathématique : rapide
  - ✓ Production de code pour : Pentium III
  
- Diagnostiques lors de la compilation
  - ✓ Niveau d'avertissement : Aucun
  - ✓ Vérification du standard Fortran : Aucun

- Vérification
  - ✓ Niveau de vérification : Minimum
- Point flottant
  - ✓ Nombre de point flottant : 3
- Optimisation
  - ✓ Niveau d'optimisation : Optimisation complète
  - ✓ But de l'optimisation : Rapidité
  - ✓ Optimisation basée sur l'architecture : Pentium III

### 3.4 Validation du module

#### 3.4.1 Modèle de simulation

Afin de valider les routines de calculs ajoutées au module « PlantCondLoopOperation.f90 », un modèle de simulation a été programmé.

##### 3.4.1.1 Bâtiment

Le bâtiment hypothétique simulé est un édifice à bureau situé à Montréal. L'édifice comporte un seul niveau d'une superficie brute de 1350m<sup>2</sup> (14526pi<sup>2</sup>). L'enveloppe du bâtiment a été choisie en fonction des dispositions obligatoires et des exigences prescriptives du code modèle national de l'énergie pour les bâtiments – Canada 1997 (CMNÉB). La composition de l'ensemble des parois opaques du bâtiment est présentée ci-dessous.

Tableau I

Caractéristiques thermodynamiques des composantes des murs extérieurs

<b>Matériaux</b>	<b><math>\rho</math></b>	<b>Cs</b>	<b><math>\epsilon</math></b>	<b><math>\alpha_s</math></b>	<b><math>\alpha_v</math></b>	<b>R</b>
Brique de façade - 4 pouces	2080	920	0.9	0.6	0.6	0.08128
Espace d'air 1/2 pouce	-	-	-	-	-	0.15900
Polystyrène extrudé - 2 pouces	29	1210	0.9	0.5	0.5	1.75172
Laine minérale - 3 pouces	9.6	710	0.9	0.75	0.75	2.10497
Espace d'air 1/2 pouce	-	-	-	-	-	0.15900
Panneau de gypse – 5/8 pouce	800	1090	0.9	0.78	0.78	0.09885

Tableau II

Caractéristiques thermodynamiques des composantes du plancher

<b>Matériaux</b>	<b><math>\rho</math></b>	<b>Cs</b>	<b><math>\epsilon</math></b>	<b><math>\alpha_s</math></b>	<b><math>\alpha_v</math></b>	<b>R</b>
Polystyrène extrudé – 1-1/2 pouces	29	1210	0.9	0.5	0.5	1.31379
Béton 2240 kg/m <sup>3</sup> - 6 pouces	2240	837	0.9	0.65	0.65	0.08811
Tuile céramique	1920	0.79	0.9	0.2	0.2	0.00980

Tableau III

Caractéristiques thermodynamiques des composantes de la toiture

<b>Matériaux</b>	<b><math>\rho</math></b>	<b>Cs</b>	<b><math>\epsilon</math></b>	<b><math>\alpha_s</math></b>	<b><math>\alpha_v</math></b>	<b>R</b>
Pierres et roches - 1/2 pouce	881	1674	0.9	0.65	0.65	0.00884
Membrane multi-couche - 3/8 pouce	1121	1674	0.9	0.75	0.75	0.05000
Polystyrène extrudé – 5 pouces	29	1210	0.9	0.5	0.5	4.37931
Béton 2240 kg/m <sup>3</sup> - 2 pouces	2240	837	0.9	0.65	0.65	0.02937

La résistance thermique globale des murs extérieurs est donc de 4.35 m<sup>2</sup>K/W comparativement à la résistance exigée par le CMNÉB qui est de 3.03 m<sup>2</sup>K/W, soit 43.6% plus élevée. La résistance thermique globale exigée par le CMNÉB est de 1.08 m<sup>2</sup>K/W pour un plancher en contact avec le sol et de 3.45 m<sup>2</sup>K/W pour une toiture. Le plancher et la toiture de l'édifice simulé ont une résistance thermique globale respective de 1.41 m<sup>2</sup>K/W (30.6% supérieure au CMNÉB) et de 4.46 m<sup>2</sup>K/W (29.3% supérieure au CMNÉB).

La fenestration de l'édifice représente 32% de la surface totale des murs extérieurs. Selon le CMNÉB la résistance thermique globale minimale du vitrage est de 0.476 m<sup>2</sup>-K/W lorsque le rapport fenêtrage-mur est inférieur à 40%. Le bâtiment simulé est donc muni d'un vitrage composé d'une vitre claire de 6mm, d'un espace d'air de 13mm et d'une vitre claire de 6mm lui conférant une résistance thermique globale de 0.515 m<sup>2</sup>-K/W. Les spécifications techniques de chaque panneau de verre sont présentées dans le tableau IV.

Tableau IV

Caractéristiques thermodynamiques des panneaux de verres

Th	Ts	Rfs	Rbs	Tv	Rfv	Rbv	Tir	Ef	Eb	Con
0.006	0.775	0.071	0.071	0.881	0.080	0.080	0.000	0.840	0.840	0.900

Lors de la modélisation, le bâtiment a dû être divisé en 9 zones réparties en 8 zones externes et 1 zone interne. Une zone est définie ici comme étant un regroupement de pièces ayant le même système de traitement de l'air (CVCA), le même type d'activité, le même profil d'occupation, la même stratégie de contrôle de la température, les mêmes caractéristiques d'enveloppe externe, une même orientation des parois extérieures et pour finir un pourcentage similaire de surfaces de vitrages. Le zonage du bâtiment est présenté à la figure 19.

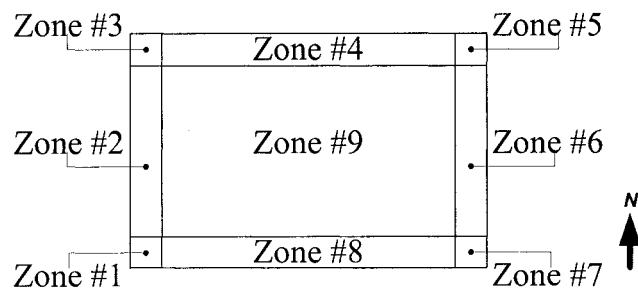


Figure 19 Zonage du bâtiment

### 3.4.1.2 Charges internes

Les caractéristiques de charges internes des zones ont été établies à partir du EE4 – Code de conformité des bâtiments par la méthode de performance. Les zones ont donc une densité d'éclairage de 19.4 W/m<sup>2</sup>, une densité d'équipement de 7.5 W/m<sup>2</sup> et un taux d'occupation d'environ 20 m<sup>2</sup>/personne. Les horaires d'exploitation sont présentées ci-dessous et sont issues du EE4.

Tableau V

## Horaires d'exploitation du bâtiment de 1h à 12h

Tableau VI

Horaires d'exploitation du bâtiment de 13h à 24h

### 3.4.1.3 Systèmes mécaniques

Pour climatiser les zones thermiques, l'édifice possède un système de traitement de l'air à volume d'air variable (VAV) et des serpentins de réchauffage terminaux à l'eau chaude. Le système VAV est muni d'un ventilateur d'alimentation développant une pression statique conforme au CMNÉB, soit 745 Pa et d'un humificateur électrique. Les zones externes sont équipées de plinthes de chauffage à l'eau chaude. La capacité de chaque plinthe a été déterminée en fonction des déperditions de chaleur de l'enveloppe du bâtiment. Les serpentins de réchauffage ont, quant à eux, la capacité d'élever la température de l'air soufflée à la température du point de consigne de pièce. Les serpentins de refroidissement et de chauffage du système VAV sont alimentés respectivement en eau refroidie et en eau chaude. Le bâtiment est équipé d'une chaudière à eau chaude qui dessert le serpentin de chauffage du système VAV, les serpentins de réchauffage terminaux et les plinthes de chauffage périphériques. On y retrouve également une tour d'eau reliée à un refroidisseur d'une capacité nominale de 81kW (23 tonnes) qui alimente en eau froide le serpentin de refroidissement du système VAV. Par ailleurs, le coefficient de performance du refroidisseur à pleine charge a été fixé à 3.8 (rendement minimal prescrit par le CMNÉB pour un groupe refroidisseur refroidi par eau et de plus de 73kW). Il est à noter que le dimensionnement des équipements a été fait dans un premier temps à partir des outils de design inclus dans EnergyPlus. Dans un deuxième temps, chaque composante du système a été revue et corrigée à l'aide des résultats de simulation de façon à optimiser l'ensemble de la mécanique du bâtiment. Le schéma de l'installation mécanique programmé dans EnergyPlus est présenté à la figure 20.

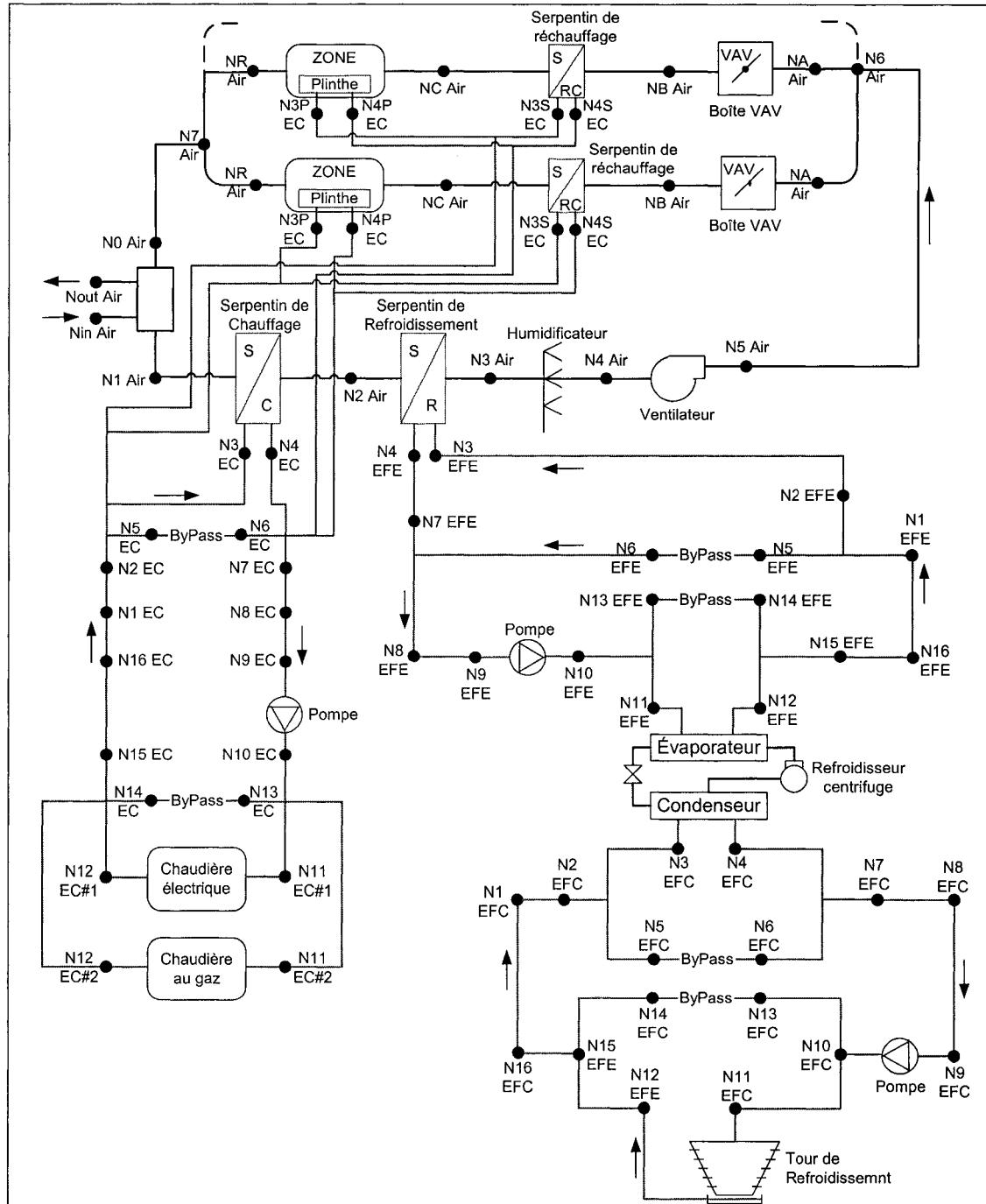


Figure 20 Schéma du système secondaire et du système primaire

En analysant la figure, on remarque la présence de deux chaudières. Ces deux chaudières peuvent être visualisées comme étant une seule chaudière munie d'un brûleur au gaz et

d'éléments électriques. Chaque chaudière a une capacité de 90 kW et la chaudière au gaz naturel a un rendement de 80% à pleine charge. Afin de limiter l'introduction de données nécessaires à la simulation des systèmes primaires, les algorithmes de calculs simplifiés d'EnergyPlus ont été privilégiés. Ainsi, l'objet « *Boiler:Simple* » a permis d'introduire les caractéristiques des deux chaudières. Parmi ces caractéristiques, on retrouve la capacité nominale, le rendement à pleine charge, la température de sortie de l'eau, le débit volumétrique d'eau, la charge partielle minimale, la charge partielle maximale et les coefficients de pondération de la consommation de combustible en fonction de la demande de chauffage. La consommation de combustible de la chaudière au gaz est calculée par EnergyPlus de la façon suivante :

$$\text{Ratio d'utilisation} = \frac{\text{Demande envoyée à la chaudière}}{\text{Capacité nominale de la chaudière}}$$

$$\text{Consommation théorique de combustible} = \frac{\text{Demande envoyée à la chaudière}}{\text{Rendement de la chaudière à pleine charge}}$$

$$\text{Consommation de combustible} = \frac{\text{Consommation théorique de combustible}}{\text{C1} + \text{C2} * \text{Ratio d'utilisation} + \text{C3} * \text{Ratio d'utilisation}^2}$$

où :

$$\text{C1} = 0.082597$$

$$\text{C2} = 0.996764$$

$$\text{C3} = -0.079361$$

La valeur de chaque coefficient (C1, C2 et C3) a été déterminée à partir du EE4 – Code de conformité des bâtiments par la méthode de performance. Évidemment, lors de la simulation de la chaudière électrique, le coefficient C1 est égal à 1 et les coefficients C2 et C3 sont nuls. L'objet « *Cooling Tower :Single Speed* » a permis de définir la capacité de la tour de refroidissement du groupe refroidisseur. On retrouve deux objets dans

EnergyPlus qui permet de simuler un refroidisseur, soit l'objet «*Chiller:Const COP*» et l'objet «*Chiller:Electric*». L'objet «*Chiller:Const COP*» permet de considérer comme constant le coefficient de performance du refroidisseur, et ce, peu importe les conditions d'opération. L'objet «*Chiller:Electric*» permet à l'utilisateur de simuler les performances réelles d'un refroidisseur en fonction de ses conditions d'opération, et ce, pour chaque pas de temps de calculs. Pour des raisons de simplification du modèle et étant donné que le but de ce modèle était d'analyser l'opération d'une chaudière biénergie, l'objet «*Chiller:Const COP*» a été privilégié. Pour finir, toutes les pompes sont à débit variable et leur pression manométrique est conforme au EE4 soit à 179.4kPa.

#### 3.4.1.4 Séquence d'opération

Le système CVCA est programmé pour démarrer à 6h et pour arrêter à 23h. La température d'alimentation du système est contrôlée en fonction de la zone la plus chaude et varie entre 12°C et 18°C. La chaudière d'eau chaude est disponible tout au long de l'année et le groupe refroidisseur est disponible du 1<sup>er</sup> avril au 1<sup>er</sup> octobre inclusivement. L'humidificateur est contrôlé via une sonde d'humidité placée dans la zone #9 dont la basse limite d'humidité relative est fixée à 30%. Les thermostats de pièces sont de type double point de consigne. Le point de consigne des zones thermiques entre 17h à 6h est de 17°C en mode chauffage et de 30°C en mode refroidissement. Entre 6h et 17h, le point de consigne de pièces est portée à 20°C en mode chauffage et à 24°C en mode refroidissement. La boîte VAV, le serpentin de réchauffage et la plinthe de chauffage sont contrôlés en séquence. Lorsque la demande en climatisation diminue, le volet de la boîte VAV restreint l'admission d'air dans la zone concernée jusqu'au minimum d'air permis. Si la température ambiante de la zone descend sous le point de consigne de chauffage, l'air soufflé dans la zone est réchauffé par le serpentin de réchauffage. Pour finir, si le serpentin de réchauffage est incapable de maintenir le point de consigne de chauffage, la plinthe de chauffage est mise en opération.

### 3.4.2 Analyse des résultats

Comme mentionné précédemment, le modèle de simulation décrit ci-haut a permis d'analyser l'opération d'une chaudière biénergie et par le fait même, de valider les séquences de contrôle du nouveau module de simulation. L'analyse des résultats de simulation démontre que le nouveau module de simulation permet de contrôler les chaudières en fonction des prescriptions d'Hydro-Québec pour la biénergie (voir la figure 21).

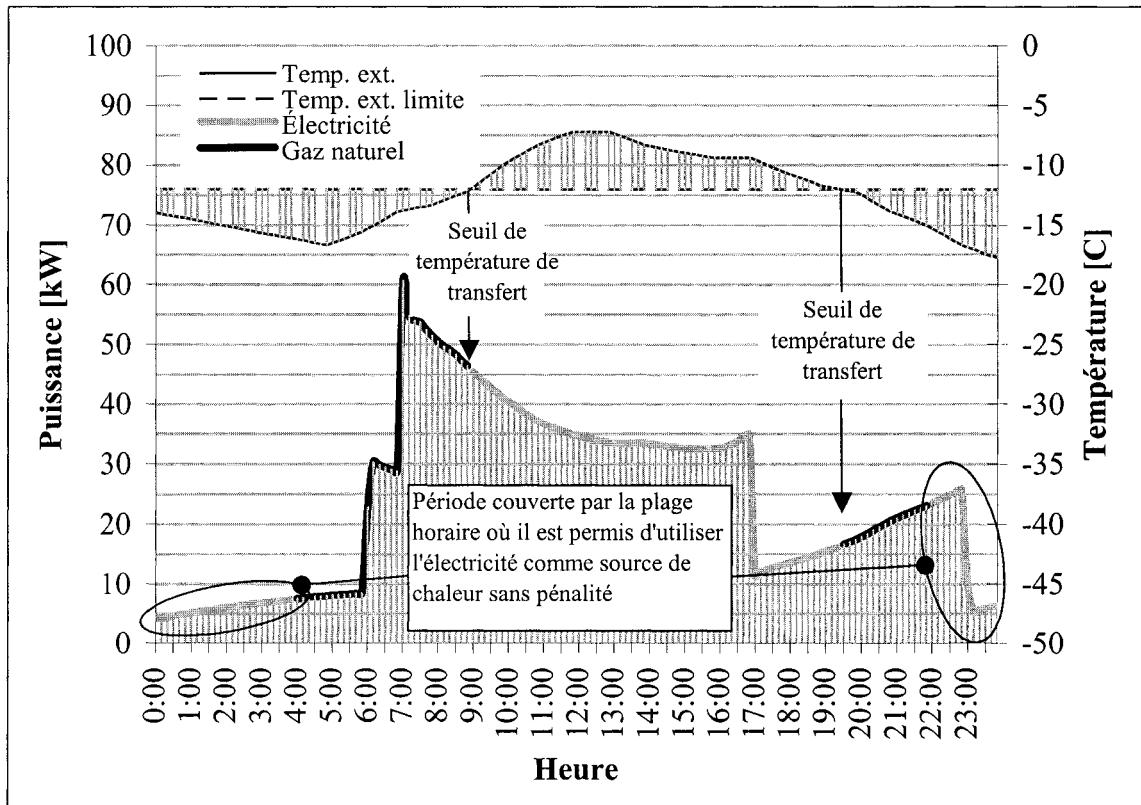


Figure 21 Fonctionnement du système biénergie

On retrouve sur la figure ci-haut la température extérieure et la puissance de chauffage horaire fournie par la chaudière le 28 janvier. Lors de la simulation, le seuil de la température de transfert (Température extérieure) a été fixé à -12°C. Même si la

température extérieure est sous le seuil de la température de transfert de minuit à 9h et de 19h30 à minuit, on remarque que la chaudière électrique est en marche de minuit à 4h et de 22h à minuit. Cette situation s'explique comme suit : selon le mode de contrôle biénergie, l'élément électrique de la chaudière a la permission de demeurer en fonction pendant la nuit soit entre 22h à 6h30 et ce, pour une plage horaire de six heures et demie.

Le module étant validé, il devenait donc intéressant d'évaluer et de comparer le coût global de l'énergie consommée par le bâtiment muni d'un système biénergie versus celui sans biénergie. Pour ce faire, trois modèles de simulation ont été programmés dans EnergyPlus. Le modèle #1 est muni d'un système biénergie utilisant l'électricité comme source d'énergie principale et le gaz naturel comme seconde source. Les modèles #2 et #3 sont équipés respectivement d'une chaudière électrique et d'une chaudière au gaz naturel. Il est à noter que l'enveloppe du bâtiment est la même pour les trois modèles et que seules les chaudières ont été modifiées. De plus, les caractéristiques de celles-ci correspondent aux caractéristiques mentionnées précédemment. Dans le but d'obtenir le maximum de précision au niveau de la demande électrique, le pas de calcul employé lors des simulations a été fixé à 10 minutes, soit le plus court qu'EnergyPlus permet d'introduire. Lors du calcul des coûts d'électricité du modèle #1, le tarif BT (Biénergie) et le tarif G (tarifs généraux de petite puissance) ont été utilisés. Le tarif G (tarifs généraux de petite puissance) a été employé lors du calcul des coûts reliés à la consommation d'électricité des modèles #2 et #3. Un prix virtuel de 0.36\$/m<sup>3</sup> de gaz naturel a permis de calculer les coûts reliés à la consommation de gaz naturel pour les modèles #1 et #3. Le détail des calculs des coûts énergétiques sont présentés en annexe. La figure 22 présente le sommaire des coûts totaux mensuels d'énergie et ce, pour chaque modèle de simulation. Il est à noter que les coûts présentés excluent toutes formes de taxes.

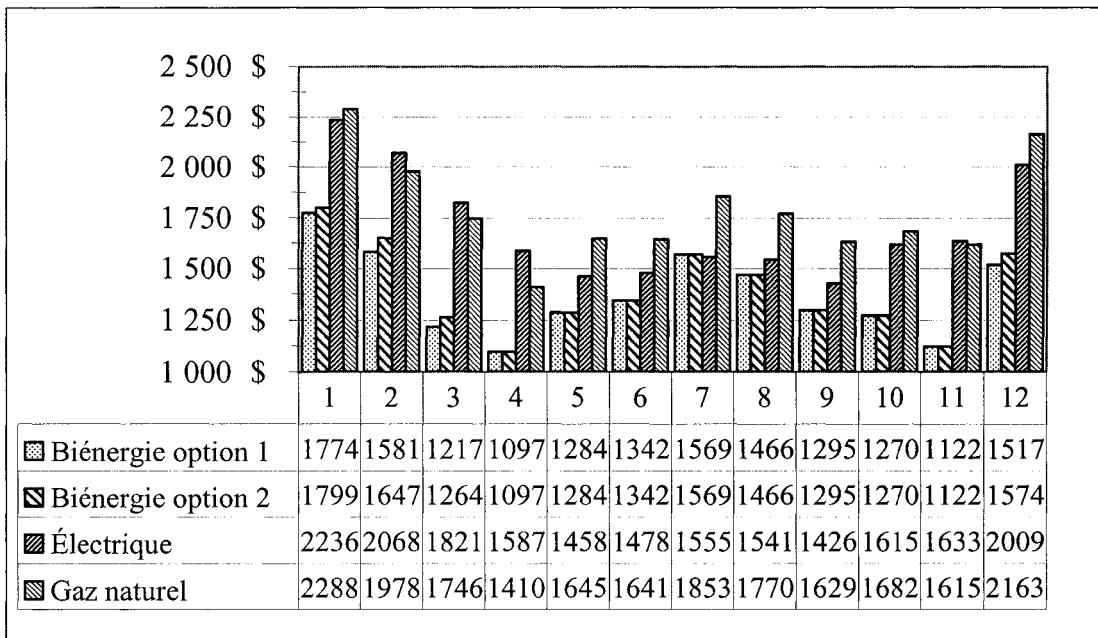


Figure 22 Coûts mensuels d'énergie

Les coûts annuels de l'énergie consommée du modèle #1 - Biénergie Option 1 (voir section 3.3.1) et du modèle #1 - Biénergie Option 2 (voir section 3.3.1) sont les plus bas de tous avec respectivement 16533.75\$ et 16729.11\$. La facture d'énergie du modèle #3 s'est avérée la plus élevée avec 21419.91\$, soit 29.6% supérieure à la facture du modèle #1 (Biénergie Option 1). Quant à la facture du modèle #2 (20427.51\$), elle est supérieure de 23.6% à la facture du modèle #1 (Biénergie Option 1).

Tel que mentionné à la section 2.5.1.5, la demande maximale des équipements est grandement influencée par le pas de temps de calculs. Étant donné qu'une partie de la facturation d'Hydro-Québec est basée sur la demande mensuelle maximale, il devenait intéressant d'évaluer l'impact du pas de temps de calcul sur les coûts liés à la consommation d'électricité. Pour ce faire, le pas de temps de calculs de tous les modèles a été augmenté de 10 minutes à 60 minutes. Le tableau comparatif des résultats obtenus est présenté ci-dessous.

Tableau VII

Influence du pas de temps de calculs sur le coût de l'électricité

Mois	Modèle #1 : Biénergie option 1			Modèle #1 : Biénergie option 2			Modèle #2 : Électricité			Modèle #3 : Gaz		
	10 min. [\$]	60 min. [\$]	Err. [%]	10 min. [\$]	60 min. [\$]	Err. [%]	10 min. [\$]	60 min. [\$]	Err. [%]	10 min. [\$]	60 min. [\$]	Err. [%]
Janv.	1 222	1 196	2.1	1 246	1 236	0.9	2 236	2 097	6.2	880	880	0.0
Fév.	1 078	1 077	0.1	1 144	1 128	1.4	2 068	1 871	9.5	775	775	0.0
Mars	1 148	1 081	5.9	1 195	1 131	5.4	1 821	1 596	12.3	853	853	0.0
Avril	1 097	1 030	6.1	1 097	1 030	6.1	1 587	1 360	14.3	876	876	0.0
Mai	1 284	1 275	0.7	1 284	1 275	0.7	1 458	1 381	5.3	1 139	1 130	0.8
Juin	1 342	1 334	0.6	1 342	1 334	0.6	1 478	1 427	3.4	1 214	1 206	0.7
Juil.	1 569	1 569	0.0	1 569	1 569	0.0	1 555	1 546	0.6	1 445	1 445	0.0
Août	1 466	1 455	0.7	1 466	1 455	0.7	1 541	1 462	5.1	1 336	1 326	0.8
Sept.	1 295	1 293	0.1	1 295	1 293	0.1	1 426	1 336	6.3	1 157	1 155	0.1
Oct.	1 270	1 270	0.0	1 270	1 270	0.0	1 615	1 426	11.7	1 108	1 108	0.0
Nov.	1 122	1 058	5.7	1 122	1 058	5.7	1 633	1 485	9.0	872	851	2.4
Déc.	1 178	1 110	5.7	1 235	1 204	2.5	2 009	1 821	9.4	827	827	0.0
<b>Total</b>	<b>15070</b>	<b>14 748</b>	<b>2.1</b>	<b>15 265</b>	<b>14 982</b>	<b>1.9</b>	<b>20 428</b>	<b>18 809</b>	<b>7.9</b>	<b>12 482</b>	<b>12 431</b>	<b>0.4</b>

On remarque que l'erreur annuelle engendrée par l'utilisation d'un pas de temps de calcul de 60 minutes au lieu de 10 minutes varie entre 0.4% et 7.9%. Le modèle #2 s'est avéré le plus affecté par le choix du pas de temps de calculs avec une différence de 12.3% en mars et 14.3% en avril. On peut donc conclure qu'il est préférable de choisir un pas de temps de calculs de 10 minutes lors d'analyses de coûts énergétiques.

## CONCLUSION

L'analyse comparative du logiciel EnergyPlus avec la base de données d'enregistrement a permis de démontrer la fiabilité des résultats obtenus et la performance des algorithmes de calculs utilisés par celui-ci. L'utilisation, due aux limites du logiciel, d'une cédule horaire de la température moyenne de soufflage a été identifié comme étant la principale source d'erreur lors de la comparaison du débit d'air soufflé dans les zones.

Lors de l'analyse des résultats obtenus avec DOE-2E et EnergyPlus, certaines différences ont été observées entre ceux-ci. Ces différences étaient principalement dues au manque de flexibilité du logiciel DOE-2. Par exemple, le pas de temps de calculs de 60 minutes du logiciel DOE-2E est fixe contrairement à celui d'EnergyPlus qui est ajustable et qui peut être réduit à 10 minutes. De plus, l'algorithme de calcul du coefficient de convection interne utilisé par défaut dans le logiciel DOE-2 peut être choisi selon la complexité de calculs dans le logiciel EnergyPlus. Il a été démontré que le choix de cet algorithme avait une influence non-négligeable sur la variation de la température ambiante des zones, et ce, surtout lorsque le système CVCA était en arrêt.

Certaines anomalies ont également été présentées. Une perte de débit massique a été observée entre certains nœuds de la boucle d'air dans EnergyPlus. Pour une journée type, la perte maximale observée était de 0.83% du débit massique total du système. De plus, l'analyse des résultats de simulation a démontré que le logiciel EnergyPlus ne respectait pas les limites fixées par l'utilisateur (débit maximum du ventilateur). Il est à noter que ces anomalies ont seulement été observées avec la version 1.0.0.023 d'EnergyPlus.

Il a été démontré que l'élaboration et l'intégration d'un module de simulation dans EnergyPlus exige une connaissance approfondie de l'architecture interne d'EnergyPlus

et du language de programmation Fortran90. Malgré les efforts faits par les développeurs d'EnergyPlus, la structure modulaire et l'intéraction entre chaque module du logiciel demeurent complexes. Le module développé dans le cadre de ce mémoire a permis de simuler avec succès le mode de contrôle biénergie.

En terminant, il a été également établi que le pas de temps de calculs avait une influence non-négligeable sur le calcul des coûts liés à la consommation d'électricité. En fait, l'augmentation du pas de temps de calculs se traduit par une diminution de la demande maximale d'électricité qui est normalement facturable par Hydro-Québec. Il deviendrait donc intéressant de développer et d'intégrer un nouveau module de simulation permettant de simuler l'impact d'un système de délestage de charge ayant pour but d'abaisser la demande maximale d'électricité d'un bâtiment qui est facturé par Hydro-Québec.

## **ANNEXE 1**

### **Module de simulation biénergie**

```

MODULE PlantCondLoopOperation

! PURPOSE OF THIS MODULE: This module selects the equipment on
! the plant and condenser loops that will operate
! for a given timestep.

! METHODOLOGY EMPLOYED: The main driver, "ManageLoopOperation"
! gets the general inputs for loop operation algorithms
! and determines whether the calling routine was the
! plant loop or the condenser loop. The plant and
! condenser loop drivers ("PlantLoopOperation and
! "CondLoopOperation") determine the type of operation
! scheme being used to sequence equipment for the
! current timestep and call the relevant utility
! routines

! REFERENCES:

! USE STATEMENTS:
USE DataGlobals, ONLY: MaxNameLength, ShowContinueError, ShowWarningError, ShowSevereError, ShowFatalError, &
BeginEnvrnFlag, BeginHourFlag, BeginSimFlag, WarmupFlag
USE DataPlant
USE DataHVACGlobals, ONLY: NumPlantLoops, NumCondLoops, FirstTimeStepSysFlag

IMPLICIT NONE      ! Enforce explicit typing of all variables

PRIVATE ! Everything private unless explicitly made public

!MODULE PARAMETER DEFINITIONS
INTEGER, PARAMETER :: HeatingOp = 1    ! Constant for Heating Operation
INTEGER, PARAMETER :: CoolingOp = 2    ! Constant for Cooling Operation
INTEGER, PARAMETER :: UncontrolledOp=3 ! Constant for Uncontrolled Operation

! DERIVED TYPE DEFINITIONS
!Variables and parameters associated with each Cond equipment operating scheme
TYPE LoopOperation
CHARACTER(len=MaxNameLength)      :: Name          ! Identifies the load based Condenser Loop operation scheme
CHARACTER(len=7)                 :: HeatOrCool
INTEGER                          :: OpType        ! Type set for this Operation List

! The following variables apply to load range based operation only
INTEGER                         :: NumLoadRangeLists   ! Number of available sets of equipment for various load ranges
CHARACTER(len=MaxNameLength), &
DIMENSION(:,),POINTER          :: EquipOperationListName ! Identifies list of available equipment for a given load range
REAL ,DIMENSION(:,),POINTER    :: LoadRangeLowerLimit ! lower limit of the load range for
'EquipOperationListName'
REAL ,DIMENSION(:,),POINTER    :: LoadRangeUpperLimit ! upper limit of the load range for 'EquipOperationListName'
END TYPE LoopOperation

!Equipment operation lists associated with Condenser Loop load based operation scheme
TYPE EquipmentLists
CHARACTER(len=40)                :: Name          ! Identifies a list of available equipment for a particular load range
INTEGER                         :: NumMachines   ! Number of available sets of equipment for given load range
CHARACTER(len=40),DIMENSION(:,),POINTER :: MachineType ! The 'KeyWord' that identifies each type of equipment
CHARACTER(len=40),DIMENSION(:,),POINTER :: MachineName ! Identifies the particular machine or type 'MachineType'
END TYPE EquipmentLists

TYPE PlantComponentMap
INTEGER :: PlantLoopNum
INTEGER :: PlantBranchNum
INTEGER :: PlantCompNum
END TYPE PlantComponentMap

TYPE CondDemandSideComponents

```

```

INTEGER :: NumDemandSideComp
TYPE(PlantComponentMap), &
    POINTER, DIMENSION(:) :: DemandSideComp ! Component type list
END TYPE CondDemandSideComponents

!MODULE VARIABLE DECLARATIONS:
! Variables required only by the "Load based operation scheme".
INTEGER, SAVE :: NumLoadRangeSchemes ! number of 'LOAD RANGE BASED OPERATION' specifications in input
INTEGER, SAVE :: NumUncontrolledSchemes ! number of 'UNCONTROLLED OPERATION BASED' specifications in input
INTEGER, SAVE :: NumEquipOperationLists ! number of 'LOAD RANGE EQUIPMENT LIST's in input
LOGICAL, SAVE :: GetOpSchemeInput = .TRUE. ! true when GetOperationScheme needs to be called
LOGICAL, SAVE :: GetPlantOpInput = .TRUE. ! true when GetPlantOperationInput needs to be called
LOGICAL, SAVE :: GetCondOpInput = .TRUE. ! true when GetCondOpInput needs to be called
INTEGER, SAVE, DIMENSION(:), ALLOCATABLE :: PlantSchemeNum() !allocate to number of plant loops
INTEGER, SAVE, DIMENSION(:), ALLOCATABLE :: CondSchemeNum() !allocate to number of cond loops
INTEGER, SAVE, DIMENSION(:), ALLOCATABLE :: PlantRangeListNum() !allocate to number of plant loops
INTEGER, SAVE, DIMENSION(:), ALLOCATABLE :: CondRangeListNum() !allocate to number of cond loops
INTEGER :: MaxNumMachines_EquipOpLists ! Maximum number of Machines on any Equipment Operation Lists

TYPE (LoopOperation), ALLOCATABLE, DIMENSION(:) :: LoadBasedScheme
TYPE (LoopOperation), ALLOCATABLE, DIMENSION(:) :: UncontrolledLoopScheme
TYPE (EquipmentLists), ALLOCATABLE, DIMENSION(:) :: EquipOperationList
TYPE(CondDemandSideComponents), ALLOCATABLE, DIMENSION(:)::MapCondLoop !allocate to number of cond. loops

```

```

!SUBROUTINE SPECIFICATIONS FOR MODULE
!Driver Routines
PUBLIC ManagePlantLoopOperation
PUBLIC ManageCondLoopOperation
!Get Input Routines
PRIVATE GetPlantOperationInput
PRIVATE GetCondOperationInput
PUBLIC GetOperationSchemeInput
!Utility Routines
PRIVATE FindLoadRangeAndEquipLists
PRIVATE TurnOnCondLoopPipes
PRIVATE TurnOnPlantLoopPipes
PRIVATE TurnOnAvailableCondEquip
PRIVATE DistributePlantLoad
PUBLIC TurnOffPlantEquipment ! Public so that the supply side manager can shut off equipment when pumps are off
PUBLIC TurnOffCondEquipment ! Public so that the supply side manager can shut off equipment when pumps are off

```

CONTAINS

! MODULE SUBROUTINES:

! Beginning of Module Driver Subroutines

\*\*\*\*\*

SUBROUTINE ManagePlantLoopOperation(LoopNum,LoopDemand,RunLoopPumps,FirstHVACIteration)

! SUBROUTINE INFORMATION:

! AUTHOR: Dan Fisher  
! DATE WRITTEN: April 1999  
! REVISED: March 2001  
! July 2001, Rick Strand (revision of pump and loop control code)

! PURPOSE OF THIS SUBROUTINE:

! ManageLoopOperation is the driver routine  
! for plant equipment selection. It calls the general "Get-  
! Input" routines, determines whether the calling routine was  
! the Plant loop manager or the condenser loop manager and  
! calls the relevant loop operation driver. "ManageLoop-  
! Operation" is called for each loop and each iteration  
! from "ManagePlantLoops" and "ManagePlantLoops" located in  
! Module "PlantLoopManager" and "PlantLoopManager" respectively.

```

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
USE ScheduleManager, ONLY: GetScheduleIndex, GetCurrentScheduleValue
USE InputProcessor, ONLY: FindItemInList

! ****
!Modification par René Bellemare (10 nov. 2002)
USE DataEnvironment, ONLY: OutDryBulbTemp
USE InputProcessor, ONLY: GetNumObjectsFound, GetObjectItem
! ****

IMPLICIT NONE ! Enforce explicit typing of all variables in this routine

! SUBROUTINE ARGUMENT DEFINITIONS:
REAL ,INTENT(IN)      :: LoopDemand
INTEGER, INTENT(IN)    :: LoopNum      ! Plant loop counter
LOGICAL, INTENT(IN)    :: RunLoopPumps
LOGICAL, INTENT(IN)    :: FirstHVACIteration

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER              :: LoadBasedSchemeNum
INTEGER              :: UncontrolledSchemeNum
INTEGER              :: Num
CHARACTER(len=MaxNameLength)  :: CurSchemeType
CHARACTER(len=MaxNameLength)  :: CurSchemeName
CHARACTER(len=20)       :: CharErrOut
INTEGER               :: LoadRangeListNum
INTEGER               :: EquipListNum
INTEGER, SAVE          :: ListErrCount
LOGICAL               :: ListError ! Equipment Operation List Error flag
LOGICAL               :: ErrorsFound = .false. ! for input errors, not saved because will only go once
LOGICAL,SAVE,ALLOCATABLE,DIMENSION(:) :: LastPumpFlag
REAL ,SAVE,ALLOCATABLE,DIMENSION(:) :: LastLoopDemand
INTEGER ,SAVE,ALLOCATABLE,DIMENSION(:) :: LastEquipListNum
LOGICAL               :: ChecklistFlag = .TRUE.
LOGICAL               :: RunLoopPumpsFlag

! ****
!Modification par René Bellemare (10 nov. 2002)
INTEGER :: OpNum
INTEGER :: NumAlphas
INTEGER :: NumNums
INTEGER :: IOSTAT
REAL, DIMENSION(30) :: NumArray
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArray

INTEGER :: OpNumBi      ! Compteur Bi-Energie
INTEGER :: NumOpBiEnergie ! Nombre total de controle BiEnergie
INTEGER :: NumAlphasBi
INTEGER :: NumNumsBi
INTEGER :: IOSTATBi

```

```

REAL, DIMENSION(30) :: NumArrayBi
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArrayBi
LOGICAL(1):: OpBiEnergieFound
INTEGER, SAVE :: ValeurBiEnergie
INTEGER :: TempBiEnergie
CHARACTER(len=MaxNameLength)      :: PlantOpScheme
! ****
! Initialize Local Variables
LoadBasedSchemeNum   = 0
LoadRangeListNum     = 0
ListError           = .FALSE.

!Get Input
IF (GetPlantOpInput)THEN
  IF (GetOpSchemeInput)THEN
    CALL GetOperationSchemeInput(ErrorsFound)
    GetOpSchemeInput = .FALSE.
  END IF
  CALL GetPlantOperationInput(ErrorsFound)
  GetPlantOpInput = .FALSE.
  IF (.not. GetCondOpInput) THEN
    CALL VerifyEquipLists
  ENDIF

  ! Initialize ON/OFF arrays to OFF
  DO Num = 1, NumPlantLoops
    CALL TurnOffPlantEquipment(Num)
  END DO
  ! IF (.NOT. ALLOCATED(LastPumpFlag) .AND. &
  !     .NOT. ALLOCATED(LastLoopDemand))THEN
  !   ALLOCATE(LastPumpFlag(NumPlantLoops))
  !   ALLOCATE(LastLoopDemand(NumPlantLoops))
  !   ALLOCATE(LastEquipListNum(NumPlantLoops))
  !   LastPumpFlag = .TRUE.
  !   LastLoopDemand = 0.0
  !   LastEquipListNum = 0
  ! END IF
  END IF

  IF (BeginHourFlag .AND. FirstHVACIteration)THEN
    ChecklistFlag = .TRUE.
  ELSE
    ChecklistFlag = .FALSE.
  END IF

  ! If nothing has changed, stick with the last equip ON-OFF array
  RunLoopPumpsFlag=(LastPumpFlag(loopnum) .and. RunLoopPumps) .or. (.not. LastPumpFlag(LoopNum) .and. .not.
  RunLoopPumps)

  IF ( (LastLoopDemand(loopnum) == loopdemand) &
       .AND. (RunLoopPumpsFlag) &
       .AND. (LastEquipListNum(LoopNum) /= 0) &
       .AND. (.NOT. ChecklistFlag) ) THEN
    CALL DistributePlantLoad(LoopNum,LastEquipListNum(LoopNum),LoopDemand)
    RETURN
  END IF

  ! Set comparators for next time.
  LastLoopDemand(loopnum) = loopdemand
  LastPumpFlag(loopnum) = RunLoopPumps

! ****
!Modification par René Bellemare (10 nov. 2002)
NumOpBiEnergie = GetNumObjectsFound('Controle BiEnergie')

```

```

! ****
! Loop through all of the specified Plant operation schemes
DO Num =1, PlantSupplySide(LoopNum)%NumOpSchemes

! ****
!Modification par René Bellemare (10 nov. 2002)
CALL GetObjectItem('PLANT OPERATION SCHEMES',LoopNum,AlphArray,NumAlphas, &
    NumArray,NumNums,IOSTAT)
PlantOpScheme = AlphArray(1)
OpBiEnergieFound = .FALSE.
DO OpNumBi = 1, NumOpBiEnergie
    CALL GetObjectItem('Controle BiEnergie',OpNumBi,AlphArrayBi,NumAlphasBi, &
        NumArrayBi, NumNumsBi, IOSTATBi)
    IF (PlantOpScheme /= AlphArrayBi(2)) CYCLE
    TempBiEnergie = NumArrayBi(1)
    IF (OutDryBulbTemp < TempBiEnergie .AND. &
        Num == PlantSupplySide(LoopNum)%NumOpSchemes) THEN
        ValeurBiEnergie = 0
        ELSE
            ValeurBiEnergie = 1
    END IF
    OpBiEnergieFound = .TRUE.
END DO
IF (.not. OpBiEnergieFound) THEN
    ValeurBiEnergie = 1
END IF

IF (GetCurrentScheduleValue(PlantSupplySide(LoopNum)%OpScheme(Num)%SchedPtr) &
    * ValeurBiEnergie <= 0.0) CYCLE

! ****
! Set local variables
CurSchemeType = PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf
CurSchemeName = PlantSupplySide(LoopNum)%OpScheme(Num)%Name

! Select First available type of Plant operation
PlantOperationScheme: SELECT CASE (CurSchemeType)

CASE ('LOAD RANGE BASED OPERATION') ! Load based controls
    ! find new load range and equipment
    CALL
FindLoadRangeAndEquipLists(LoopDemand,CurSchemeName,EquipListNum,LoadBasedSchemeNum,LoadRangeListNum,ListER
ROR)
LastEquipListNum(LoopNum) = EquipListNum
    ! error check
    IF (EquipListNum > 0)THEN
        ! Initilaize all equipment to OFF before turning on the available equipment. This always leaves the pumps alone.
        CALL TurnOffPlantEquipment(LoopNum)
        ! Turn ON pipes and available equipment only!
        CALL TurnOnPlantLoopPipes(LoopNum)
        CALL DistributePlantLoad(LoopNum,EquipListNum,LoopDemand)
    ELSE
        CALL TurnOffPlantEquipment(LoopNum)
    END IF

CASE ('UNCONTROLLED LOOP OPERATION') ! Uncontrolled loop
    ! Find uncontrolled scheme number -- validation happens earlier (I hope -- LKL)
    UncontrolledSchemeNum = FindItemInList(CurSchemeName, UncontrolledLoopScheme%Name, NumUncontrolledSchemes)

    ! Find corresponding equipment list number
    EquipListNum = FindItemInList(UncontrolledLoopScheme(UncontrolledSchemeNum)%EquipOperationListName(1), &
        EquipOperationList%Name, NumEquipOperationLists)
LastEquipListNum(LoopNum) = EquipListNum
    ! error check

```

```

IF (EquipListNum > 0)THEN
  ! Initilaize all equipment to OFF before turning on the available equipment. This always leaves the pumps alone.
  CALL TurnOffPlantEquipment(LoopNum)
  ! Turn ON pipes and available equipment only!
  CALL TurnOnPlantLoopPipes(LoopNum)
  CALL DistributePlantLoad(LoopNum,EquipListNum,LoopDemand)
ELSE
  ListError = .TRUE.
  CALL TurnOffPlantEquipment(LoopNum)
END IF

CASE DEFAULT

  ! This is a fatal error
  CALL ShowFatalError('Invalid Operation Scheme Type Requested='//TRIM(CurSchemeType)// &
    ', in ManagePlantLoopOperation')

END SELECT PlantOperationScheme

  ! Error handling

  ! No valid equipment list specified
IF (ListError) THEN
  ListErrCount=ListErrCount+1
  IF (ListErrCount < 15) THEN
    CALL ShowWarningError ('No equipment list specified in ManagePlantLoopOperation--all equipment turned off')
  ELSE
    IF (MOD(ListErrCount,100) == 0) THEN
      WRITE(CharErrOut,*) ListErrCount
      CharErrOut=ADJUSTL(CharErrOut)
      CALL ShowWarningError ('No equipment list specified in ManagePlantLoopOperation--all equipment turned off continues...'&
        //CharErrOut)
    END IF
  END IF
END IF

END DO !Loop through all of the specified Plant operation schemes

RETURN
END SUBROUTINE ManagePlantLoopOperation
*****SUBROUTINE ManageCondLoopOperation(LoopNum, LoopDemand, RunLoopPumps, FirstHVACIteration)
! SUBROUTINE INFORMATION:
!   AUTHOR: Dan Fisher
!   DATE WRITTEN: April 1999
!   REVISED: March 2001
!           July 2001, Rick Strand (revision of pump and loop control code)

! PURPOSE OF THIS SUBROUTINE:
! ManageLoopOperation is the driver routine
! for plant equipment selection. It calls the general "Get-
! Input" routines, determines whether the calling routine was
! the Plant loop manager or the condenser loop manager and
! calls the relevant loop operation driver. "ManageLoop-
! Operation" is called for each loop and each iteration
! from "ManagePlantLoops" and "ManageCondLoops" located in
! Module "PlantLoopManager" and "CondLoopManager" respectively.

! METHODOLOGY EMPLOYED:

! REFERENCES:

```

```

! USE STATEMENTS:
USE ScheduleManager, ONLY: GetScheduleIndex, GetCurrentScheduleValue
USE InputProcessor, ONLY: FindItemInList

IMPLICIT NONE ! Enforce explicit typing of all variables in this routine

! SUBROUTINE ARGUMENT DEFINITIONS:
REAL ,INTENT(IN)      :: LoopDemand
LOGICAL, INTENT(IN)    :: RunLoopPumps
INTEGER, INTENT(IN)    :: LoopNum      ! Plant loop counter
LOGICAL, INTENT(IN)    :: FirstHVACIteration

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER              :: LoadBasedSchemeNum
INTEGER              :: UncontrolledSchemeNum
INTEGER              :: Num
CHARACTER(len=MaxNameLength)  :: CurSchemeType
CHARACTER(len=MaxNameLength)  :: CurSchemeName
CHARACTER(len=20)        :: CharErrOut
INTEGER              :: LoadRangeListNum
INTEGER              :: EquipListNum
INTEGER, SAVE          :: ListErrCount
LOGICAL               :: ListError ! Equipment Operation List Error flag
LOGICAL               :: ErrorsFound = .false. ! for input errors, not saved because will only go once
LOGICAL,SAVE,ALLOCATABLE,DIMENSION(:) :: LastPumpFlag
REAL ,SAVE,ALLOCATABLE,DIMENSION(:) :: LastLoopDemand
INTEGER ,SAVE,ALLOCATABLE,DIMENSION(:) :: LastEquipListNum
LOGICAL               :: ChecklistFlag = .TRUE.
LOGICAL               :: RunLoopPumpsFlag

!Initialize Local Variables
LoadBasedSchemeNum = 0
LoadRangeListNum   = 0
ListError         = .FALSE.

!Get Input
IF (GetCondOpInput)THEN
  IF (GetOpSchemeInput)THEN ! Get these first so can verify
    CALL GetOperationSchemeInput(ErrorsFound)
    GetOpSchemeInput = .FALSE.
  END IF
  CALL GetCondOperationInput(ErrorsFound)
  GetCondOpInput = .FALSE.
  IF (.not. GetPlantOpInput) THEN
    CALL VerifyEquipLists
  ENDIF

  ! Initialize ON/OFF arrays to OFF
  DO Num = 1, NumCondLoops
    CALL TurnOffCondEquipment(Num)
  END DO

  ! IF (.NOT. ALLOCATED(LastPumpFlag) .AND. &
  !     .NOT. ALLOCATED(LastLoopDemand))THEN
  !   ALLOCATE(LastPumpFlag(NumCondLoops))
  !   ALLOCATE(LastLoopDemand(NumCondLoops))

```

```

ALLOCATE(LastEquipListNum(NumCondLoops))
LastPumpFlag = .TRUE.
LastLoopDemand = 0.0
LastEquipListNum = 0
! END IF
END IF

IF (BeginHourFlag .AND. FirstHVACIteration)THEN
ChecklistFlag = .TRUE.
ELSE
ChecklistFlag = .FALSE.
END IF

! If nothing has changed, stick with the last equip ON-OFF array
RunLoopPumpsFlag=(LastPumpFlag(loopnum) .and. RunLoopPumps) .or. (.not. LastPumpFlag(LoopNum) .and. .not.
RunLoopPumps)

IF ( (LastLoopDemand(loopnum) == loopdemand) &
.AND. (RunLoopPumpsFlag) &
.AND. (LastEquipListNum(LoopNum) /= 0) &
.AND. .NOT. ChecklistFlag ) THEN
RETURN
END IF

! Set comparators for next time.
LastLoopDemand(loopnum) = loopdemand
LastPumpFlag(loopnum) = RunLoopPumps

!Loop through all of the specified Cond operation schemes
DO Num =1, CondSupplySide(LoopNum)%NumOpSchemes

! If the operating scheme is scheduled "OFF", go to next scheme
IF(GetCurrentScheduleValue(CondSupplySide(LoopNum)%OpScheme(Num)%SchedPtr) <= 0.0) CYCLE

! Set local variables
CurSchemeType = CondSupplySide(LoopNum)%OpScheme(Num)%TypeOf
CurSchemeName = CondSupplySide(LoopNum)%OpScheme(Num)%Name

! Select First available type of Cond operation
CondOperationScheme: SELECT CASE (CurSchemeType)

CASE ('LOAD RANGE BASED OPERATION') ! Load based controls
    ! find new load range and equipment
    CALL
FindLoadRangeAndEquipLists(LoopDemand,CurSchemeName,EquipListNum,LoadBasedSchemeNum,LoadRangeListNum,ListER
ROR)
    LastEquipListNum(LoopNum) = EquipListNum
    ! error check
    IF (EquipListNum > 0) THEN
        ! First turn OFF all of the equipment
        CALL TurnOffCondEquipment(LoopNum)
        ! Then Turn ON pipes and available equipment
        CALL TurnOnCondLoopPipes(LoopNum)
        CALL TurnOnAvailableCondEquip(LoopNum,EquipListNum)
    ELSE
        CALL TurnOffCondEquipment(LoopNum)
    END IF

CASE ('UNCONTROLLED LOOP OPERATION') ! Uncontrolled loop
    ! Find uncontrolled scheme number -- validation happens earlier (I hope -- LKL)
    UncontrolledSchemeNum = FindItemInList(CurSchemeName, UncontrolledLoopScheme%Name, NumUncontrolledSchemes)

    ! Find corresponding equipment list number
    EquipListNum = FindItemInList(UncontrolledLoopScheme(UncontrolledSchemeNum)%EquipOperationListName(1), &
        EquipOperationList%Name, NumEquipOperationLists)

```

```

LastEquipListNum(LoopNum) = EquipListNum
    ! error check
IF (EquipListNum > 0) THEN
    ! First turn OFF all of the equipment
    CALL TurnOffCondEquipment(LoopNum)
    ! Then Turn ON pipes and available equipment
    CALL TurnOnCondLoopPipes(LoopNum)
    CALL TurnOnAvailableCondEquip(LoopNum,EquipListNum)
ELSE
    ListError = .TRUE.
    CALL TurnOffCondEquipment(LoopNum)
END IF

CASE DEFAULT
    ! This is a fatal error
    CALL ShowFatalError('Invalid Operation Scheme Type Requested='//TRIM(CurSchemeType)// &
        ', in ManageCondLoopOperation')

END SELECT CondOperationScheme

! Error handling

    ! No valid equipment list specified
IF ( ListError)THEN
    CALL TurnOffCondEquipment(LoopNum)
    ListErrCount=ListErrCount+1
    IF (ListErrCount < 15) THEN
        CALL ShowWarningError ('No equipment list specified in ManageCondLoopOperation--all equipment turned off')
    ELSE
        IF (MOD(ListErrCount,100) == 0) THEN
            WRITE(CharErrOut,*) ListErrCount
            CharErrOut=ADJUSTL(CharErrOut)
            CALL ShowWarningError ('No equipment list specified in ManageCondLoopOperation--all equipment turned off continues='
&
                //CharErrOut)
        END IF
    END IF
END IF

END DO !Loop through all of the specified Cond operation schemes

RETURN

END SUBROUTINE ManageCondLoopOperation

! Beginning of Module GetInput subroutines
!*****SUBROUTINE INFORMATION*****
!*****SUBROUTINE GetPlantOperationInput(ErrorsFound)

SUBROUTINE GetPlantOperationInput(ErrorsFound)

    ! SUBROUTINE INFORMATION:
    !   AUTHOR      Dan Fisher
    !   DATE WRITTEN October 1998
    !   MODIFIED    na
    !   RE-ENGINEERED na

    ! PURPOSE OF THIS SUBROUTINE: This subroutine reads the primary plant loop
    ! operation schemes from the input file

    ! METHODOLOGY EMPLOYED: calls the Input Processor to retrieve data from input file.
    ! The format of the Energy+.idd (the EnergyPlus input data dictionary) for the
    ! following keywords is reflected exactly in this subroutine:
    !   LOAD RANGE BASED OPERATION
    !   LOAD RANGE EQUIPMENT LIST

    ! REFERENCES:

```

```

! na

! USE STATEMENTS:
USE ScheduleManager, ONLY: GetScheduleIndex
USE InputProcessor, ONLY: GetNumObjectsFound, GetObjectItem, FindItemInList

! ****
!Modification par René Bellemare (10 nov. 2002)
! USE DataGlobals, ONLY: ShowWarningError, ShowSevereError, ShowFatalError
! ****

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
LOGICAL, INTENT(INOUT) :: ErrorsFound ! Passed in from OpSchemeInput

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER :: LoopNum      ! Loop counter
INTEGER :: OpNum        ! Scheme counter
INTEGER :: Num          ! Item counter
INTEGER :: NumPlantLoops ! Total Number of Plant Loops
INTEGER :: NumOpSchemes ! Total Number of Operation Schemes
INTEGER :: NumAlphas
INTEGER :: NumNums
INTEGER :: IOSTAT
REAL, DIMENSION(30) :: NumArray
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArray
LOGICAL(1):: OpSchemeFound
CHARACTER(len=MaxNameLength)      :: PlantOpScheme
INTEGER :: Scheme

! ****
!Modification par René Bellemare (10 nov. 2002)
INTEGER :: OpNumBi      ! Compteur Bi-Energie
INTEGER :: NumOpBiEnergie ! Nombre total de controle BiEnergie
INTEGER :: NumAlphasBi
INTEGER :: NumNumsBi
INTEGER :: IOSTATBi
REAL, DIMENSION(30) :: NumArrayBi
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArrayBi
LOGICAL(1):: OpBiEnergieFound
INTEGER, SAVE :: ValeurBiEnergie
INTEGER :: TempBiEnergie
! ****

!Get the number of primary plant loops and operation schemes
NumPlantLoops = GetNumObjectsFound('PLANT LOOP')
NumOpSchemes = GetNumObjectsFound('PLANT OPERATION SCHEMES')

! ****
!Modification par René Bellemare (10 nov. 2002)
NumOpBiEnergie = GetNumObjectsFound('Controle BiEnergie')
! ****

!ALLOCATE THE ARRAYS IN THE LOOPCOMPONENTS TYPE TO THE
!NUMBER OF PLANT AND CONDENSER LOOPS AND FILL ARRAYS

```

```

IF (NumPlantLoops > 0)THEN
  ALLOCATE (PlantSchemeNum(NumPlantLoops))
  ALLOCATE (PlantRangeListNum(NumPlantLoops))
  IF(NOT. ALLOCATED(PlantSupplySide).AND. &
    .NOT. ALLOCATED(PlantDemandSide)) THEN
    ALLOCATE (PlantSupplySide(NumPlantLoops))
    ALLOCATE (PlantDemandSide(NumPlantLoops))
  ENDIF
ENDIF

DO LoopNum = 1, NumPlantLoops
  PlantOpScheme = PlantSupplySide(LoopNum)%OperationScheme
  OpSchemeFound=.false.

! *****
!Modification par René Bellemare (10 nov. 2002)
  OpBiEnergieFound=.FALSE.
! *****

  DO OpNum = 1, NumOpSchemes
    CALL GetObjectItem('PLANT OPERATION SCHEMES',OpNum,AlphArray,NumAlphas, &
      NumArray,NumNums,IOSTAT)
    IF (PlantOpScheme /= AlphArray(1)) CYCLE
    OpSchemeFound=.true.

! *****
!Modification par René Bellemare (10 nov. 2002)
    DO OpNumBi = 1, NumOpBiEnergie
      CALL GetObjectItem('Controle BiEnergie',OpNumBi,AlphArrayBi,NumAlphasBi, &
        NumArrayBi, NumNumsBi, IOSTATBi)
      IF (PlantOpScheme /= AlphArrayBi(2)) CYCLE
      OpBiEnergieFound=.TRUE.
    END DO

    IF (.Not. OpBiEnergieFound) THEN
      IF (NumOpBiEnergie >= 1) THEN
        IF (NumOpSchemes == OpNum) THEN
          CALL ShowSevereError('Controle Bi-Energie sur='//TRIM(AlphArrayBi(2)))
          ErrorsFound=.true.
        END IF
      END IF

      IF (OpBiEnergieFound) THEN
        CALL ShowWarningError ('Controle Bi-Energie sur='//TRIM(PlantOpScheme))
      END IF
    END IF
! *****

    PlantSupplySide(LoopNum)%NumOpSchemes = (NumAlphas - 1)/3
    ALLOCATE (PlantSupplySide(LoopNum)%OpScheme(PlantSupplySide(LoopNum)%NumOpSchemes))
    DO Num = 1, PlantSupplySide(LoopNum)%NumOpSchemes
      PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf = AlphArray(Num*3-1)
      IF (PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf /= 'LOAD RANGE BASED OPERATION'.and. &
        PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf /= 'UNCONTROLLED LOOP OPERATION') THEN
        CALL ShowSevereError('Invalid Plant Operation
Scheme='//TRIM(PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf)// &
          ', entered for Plant Scheme='//TRIM(PlantOpScheme))
        ErrorsFound=.true.
      ENDIF
      PlantSupplySide(LoopNum)%OpScheme(Num)%Name = AlphArray(Num*3)
      IF (PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf == 'LOAD RANGE BASED OPERATION') THEN
        Scheme=FindItemInList(PlantSupplySide(LoopNum)%OpScheme(Num)%Name,LoadBasedScheme%Name,NumLoadRangeSchem
es)
        IF (Scheme == 0) THEN

```

```

CALL ShowSevereError('Invalid Scheme Name Entered://'TRIM(PlantSupplySide(LoopNum)%OpScheme(Num)%Name)//
&
      ', in Plant Operation Scheme://'TRIM(PlantOpScheme))
ErrorsFound=.true.
ENDIF
ELSEIF (PlantSupplySide(LoopNum)%OpScheme(Num)%TypeOf == 'UNCONTROLLED LOOP OPERATION') THEN
Scheme=FindItemInList(PlantSupplySide(LoopNum)%OpScheme(Num)%Name,UncontrolledLoopScheme%Name,NumUncontroll
edSchemes)
IF (Scheme == 0) THEN
  CALL ShowSevereError('Invalid Scheme Name Entered://'TRIM(PlantSupplySide(LoopNum)%OpScheme(Num)%Name)//
&
      ', in Plant Operation Scheme://'TRIM(PlantOpScheme))
  ErrorsFound=.true.
ENDIF
ENDIF
PlantSupplySide(LoopNum)%OpScheme(Num)%Sched = AlphArray(Num*3+1)

PlantSupplySide(LoopNum)%OpScheme(Num)%SchedPtr=GetScheduleIndex(PlantSupplySide(LoopNum)%OpScheme(Num)%Sc
hed)
IF (PlantSupplySide(LoopNum)%OpScheme(Num)%SchedPtr == 0) THEN
  CALL ShowSevereError('Plant Operation Schedule not
found://'TRIM(PlantSupplySide(LoopNum)%OpScheme(Num)%Sched)// &
      'for Plant Operation Scheme://'TRIM(PlantOpScheme))
  Errorsfound=.true.
ENDIF
END DO
END DO
IF (.not. OpSchemeFound) THEN
  CALL ShowSevereError('Plant Operation Scheme not found://'TRIM(PlantOpScheme))
  Errorsfound=.true.
ENDIF
END DO

IF (Errorsfound) THEN
  CALL ShowFatalError('Errors found in getting Plant Operation Input')
ENDIF

RETURN

END SUBROUTINE GetPlantOperationInput

SUBROUTINE GetCondOperationInput(ErrorsFound)

! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN October 1998
!   MODIFIED    August 2001, LKL -- Validations
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE:
! This subroutine reads the primary Cond loop
! operation schemes from the input file

! METHODOLOGY EMPLOYED:
! calls the Input Processor to retrieve data from input file.
! The format of the Energy+.idd (the EnergyPlus input data dictionary) for the
! following keywords is reflected exactly in this subroutine:
!   LOAD RANGE BASED OPERATION
!   LOAD RANGE EQUIPMENT LIST

! REFERENCES:
! na

! USE STATEMENTS:

```

```

USE ScheduleManager, ONLY: GetScheduleIndex
USE InputProcessor, ONLY: GetNumObjectsFound, GetObjectItem

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
LOGICAL, INTENT(INOUT) :: ErrorsFound ! Passed in from OpSchemeInput

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER :: LoopNum      ! Loop counter
INTEGER :: OpNum        ! Scheme counter
INTEGER :: Num          ! Item counter
INTEGER :: NumCondLoops ! Total Number of Cond Loops
INTEGER :: NumOpSchemes ! Total Number of Operation Schemes
INTEGER :: NumAlphas
INTEGER :: NumNums
INTEGER :: IOSTAT
REAL, DIMENSION(30) :: NumArray
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArray
CHARACTER(len=MaxNameLength) :: CondOpScheme

!Get the number of primary Cond loops and operation schemes
NumCondLoops = GetNumObjectsFound('CONDENSER LOOP')
NumOpSchemes = GetNumObjectsFound('CONDENSER OPERATION SCHEMES')

!ALLOCATE THE ARRAYS IN THE LOOPCOMPONENTS TYPE TO THE
!NUMBER OF Cond AND CONDENSER LOOPS AND FILL ARRAYS

IF (NumCondLoops > 0)THEN
  ALLOCATE (CondSchemeNum(NumCondLoops))
  ALLOCATE (CondRangeListNum(NumCondLoops))
  IF(.NOT. ALLOCATED(CondSupplySide) .AND. &
     .NOT. ALLOCATED(CondDemandSide)) THEN
    ALLOCATE (CondSupplySide(NumCondLoops))
    ALLOCATE (CondDemandSide(NumCondLoops))
  ENDIF
ENDIF

! Make Sure There is a One to

DO LoopNum = 1, NumCondLoops

  CondOpScheme = CondSupplySide(LoopNum)%OperationScheme
  DO OpNum = 1, NumOpSchemes
    CALL GetObjectItem('CONDENSER OPERATION SCHEMES',OpNum,AlphArray,NumAlphas, &
                      NumArray,NumNums,IOSTAT)
    IF (CondOpScheme /= AlphArray(1)) CYCLE
    CondSupplySide(LoopNum)%NumOpSchemes = (NumAlphas - 1)/3
    ALLOCATE (CondSupplySide(LoopNum)%OpScheme(CondSupplySide(LoopNum)%NumOpSchemes))
    DO Num = 1, CondSupplySide(LoopNum)%NumOpSchemes
      CondSupplySide(LoopNum)%OpScheme(Num)%TypeOf = AlphArray(Num*3-1)
      CondSupplySide(LoopNum)%OpScheme(Num)%Name = AlphArray(Num*3)
      CondSupplySide(LoopNum)%OpScheme(Num)%Sched = AlphArray(Num*3+1)
    CondSupplySide(LoopNum)%OpScheme(Num)%SchedPtr=GetScheduleIndex(CondSupplySide(LoopNum)%OpScheme(Num)%Sc
    hed)
    IF (CondSupplySide(LoopNum)%OpScheme(Num)%SchedPtr == 0) THEN

```

```

    CALL ShowSevereError('Condenser Operation Schedule not
found://'TRIM(CondSupplySide(LoopNum)%OpScheme(Num)%Sched)// &
'for Condenser Operation Scheme://'TRIM(CondOpScheme))
    Errorsfound=.true.
  ENDIF
END DO
END DO
END DO

IF (Errorsfound) THEN
  CALL ShowFatalError('Errors found in getting Condenser Operation Input')
ENDIF

RETURN

END SUBROUTINE GetCondOperationInput

SUBROUTINE GetOperationSchemeInput(ErrorsFound)

! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN October 1998
!   MODIFIED    August 2001, LKL -- Validations
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE:
! This subroutine reads the primary plant loop
! operation schemes from the input file

! METHODOLOGY EMPLOYED:
! calls the Input Processor to retrieve data from input file.
! The format of the Energy+.idd (the EnergyPlus input data dictionary) for the
! following keywords is reflected exactly in this subroutine:
!   LOAD RANGE BASED OPERATION
!   LOAD RANGE EQUIPMENT LIST

! REFERENCES:
! na

! USE STATEMENTS:
USE InputProcessor, ONLY: GetNumObjectsFound, GetObjectItem, VerifyName, FindItemInList

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
LOGICAL, INTENT(INOUT) :: ErrorsFound ! May be set here and passed on

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER :: SchemeNum
INTEGER :: ListNum
INTEGER :: Num
INTEGER :: MachineNum
INTEGER :: NumAlphas
INTEGER :: NumNums
INTEGER :: IOSTAT
REAL, DIMENSION(30) :: NumArray
CHARACTER(len=MaxNameLength), DIMENSION(30) :: AlphArray

```

```

LOGICAL :: IsNotOK      ! Flag to verify name
LOGICAL :: IsBlank       ! Flag for blank name
INTEGER :: LRBO ! Plain Load Range Based Operation Inputs
INTEGER :: CLRBO ! Cooling Load Range Based Operation Inputs
INTEGER :: HLRBO ! Heating Load Range Based Operation Inputs
INTEGER :: Count ! For processing the Load Range Based Operation Inputs
LOGICAL :: PosHtgType ! Possible Heating Type for Load Range Based Operation
INTEGER :: EquipListNum

! All the Controls are loaded into the same derived type, both the PI and Limit
! These Controls are separate objects and loaded sequentially, but will
! be retrieved by name as they are needed.
LRBO = GetNumObjectsFound('LOAD RANGE BASED OPERATION')
CLRBO = GetNumObjectsFound('COOLING LOAD RANGE BASED OPERATION')
HLRBO = GetNumObjectsFound('HEATING LOAD RANGE BASED OPERATION')
NumLoadRangeSchemes=LRBO+CLRBO+HLRBO
NumUncontrolledSchemes = GetNumObjectsFound('UNCONTROLLED LOOP OPERATION')
IF (NumLoadRangeSchemes + NumUncontrolledSchemes.le.0) THEN
  CALL ShowFatalError('No Plant/Cond Loop Operation Controls Specified. Stop Simulation')
ENDIF
ALLOCATE(LoadBasedScheme(NumLoadRangeSchemes))
LoadBasedScheme%Name=''
LoadBasedScheme%HeatOrCool=''
LoadBasedScheme%OpType=0
LoadBasedScheme%NumLoadRangeLists=0
ALLOCATE(UncontrolledLoopScheme(NumUncontrolledSchemes))
UncontrolledLoopScheme%Name=''
UncontrolledLoopScheme%HeatOrCool=''
UncontrolledLoopScheme%OpType=UncontrolledOp
UncontrolledLoopScheme%NumLoadRangeLists=1

! Load all the Uncontrolled loop operation schemes into derived type
DO SchemeNum = 1, NumUncontrolledSchemes
  CALL GetObjectItem('UNCONTROLLED LOOP OPERATION',SchemeNum, &
    AlphArray,NumAlphas,NumArray,NumNums,IOSTAT)
  IsNotOK=.false.
  IsBlank=.false.
  CALL VerifyName(AlphArray(1),UncontrolledLoopScheme%Name,SchemeNum-1,IsNotOK,IsBlank,'UNCONTROLLED
  LOOP OPERATION Name')
  IF (IsNotOK) THEN
    ErrorsFound=.true.
    IF (IsBlank) AlphArray(1)='xxxxx'
  ENDIF
  ALLOCATE (UncontrolledLoopScheme(SchemeNum)%EquipOperationListName(1))
  ALLOCATE (UncontrolledLoopScheme(SchemeNum)%LoadRangeUpperLimit(0))
  ALLOCATE (UncontrolledLoopScheme(SchemeNum)%LoadRangeLowerLimit(0))
  !after allocating them assign values
  UncontrolledLoopScheme(SchemeNum)%Name      = AlphArray(1)
  UncontrolledLoopScheme(SchemeNum)%EquipOperationListName(1) = AlphArray(2)

END DO

! Load all the Load Range based loop operation schemes into derived type
Count=0
DO SchemeNum = 1, LRBO
  CALL GetObjectItem('LOAD RANGE BASED OPERATION',SchemeNum, &
    AlphArray,NumAlphas,NumArray,NumNums,IOSTAT)
  IsNotOK=.false.
  IsBlank=.false.
  CALL VerifyName(AlphArray(1),LoadBasedScheme%Name,Count,IsNotOK,IsBlank,'LOAD RANGE BASED OPERATION
  Name')
  IF (IsNotOK) THEN
    ErrorsFound=.true.
    IF (IsBlank) AlphArray(1)='xxxxx'
  ENDIF
  Count=Count+1

```

```

LoadBasedScheme(Count)%Name      = AlphArray(1)

LoadBasedScheme(Count)%NumLoadRangeLists = NumAlphas - 1
IF (LoadBasedScheme(Count)%NumLoadRangeLists <= 0) THEN
  CALL ShowSevereError('Priority Plant Control specified without equipment list, for scheme='//TRIM(AlphArray(1)))
  ErrorsFound=.true.
  CYCLE
END IF

ALLOCATE (LoadBasedScheme(Count)%EquipOperationListName(LoadBasedScheme(Count)%NumLoadRangeLists))
ALLOCATE (LoadBasedScheme(Count)%LoadRangeUpperLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
ALLOCATE (LoadBasedScheme(Count)%LoadRangeLowerLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
DO ListNum = 1, LoadBasedScheme(Count)%NumLoadRangeLists
  LoadBasedScheme(Count)%LoadRangeLowerLimit(ListNum) = NumArray(ListNum*2-1)
  LoadBasedScheme(Count)%LoadRangeUpperLimit(ListNum) = NumArray(ListNum*2)
  LoadBasedScheme(Count)%EquipOperationListName(ListNum) = AlphArray(ListNum+1)
END DO

! Determine Heating/Cooling Type
IF (ANY(LoadBasedScheme(Count)%LoadRangeLowerLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) > 0.0) .or. &
  ANY(LoadBasedScheme(Count)%LoadRangeUpperLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) > 0.0)) THEN
  PosHtgType=.true.
ELSE
  PosHtgType=.false.
ENDIF
IF (PosHtgType) THEN
  IF (ANY(LoadBasedScheme(Count)%LoadRangeLowerLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) < 0.0) .or.
  &
  ANY(LoadBasedScheme(Count)%LoadRangeUpperLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) < 0.0))
THEN
  CALL ShowSevereError('Load Range Based Operation Scheme cannot have mixed pos/neg ranges, found in scheme=' &
    //TRIM(AlphArray(1)))
  ErrorsFound=.true.
ENDIF
ELSE
  IF (ANY(LoadBasedScheme(Count)%LoadRangeLowerLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) > 0.0) .or.
  &
  ANY(LoadBasedScheme(Count)%LoadRangeUpperLimit(1:LoadBasedScheme(Count)%NumLoadRangeLists) > 0.0))
THEN
  CALL ShowSevereError('Load Range Based Operation Scheme cannot have mixed pos/neg ranges, found in scheme=' &
    //TRIM(AlphArray(1)))
  ErrorsFound=.true.
ENDIF
ENDIF
IF (PosHtgType) THEN
  LoadBasedScheme(Count)%OpType=HeatingOp
  LoadBasedScheme(Count)%HeatorCool='HEATING'
ELSE
  LoadBasedScheme(Count)%OpType=CoolingOp
  LoadBasedScheme(Count)%HeatorCool='COOLING'
ENDIF
DO ListNum = 1, LoadBasedScheme(Count)%NumLoadRangeLists
  LoadBasedScheme(Count)%LoadRangeLowerLimit(ListNum) =
ABS(LoadBasedScheme(Count)%LoadRangeLowerLimit(ListNum))
  LoadBasedScheme(Count)%LoadRangeUpperLimit(ListNum) =
ABS(LoadBasedScheme(Count)%LoadRangeUpperLimit(ListNum))
END DO
END DO

DO SchemeNum = 1, CLRBO
  CALL GetObjectItem('COOLING LOAD RANGE BASED OPERATION',SchemeNum, &
    AlphArray,NumAlphas,NumArray,NumNums,IOSTAT)
  IsNotOK=.false.
  IsBlank=.false.
  CALL VerifyName(AlphArray(1),LoadBasedScheme%Name,Count,IsNotOK,IsBlank,'LOAD RANGE BASED OPERATION
Name')
  IF (IsNotOK) THEN
    ErrorsFound=.true.

```

```

        IF (IsBlank) AlphArray(1)='xxxxx'
        ENDIF
        Count=Count+1
        LoadBasedScheme(Count)%Name      = AlphArray(1)

        LoadBasedScheme(Count)%NumLoadRangeLists = NumAlphas - 1
        IF (LoadBasedScheme(Count)%NumLoadRangeLists <= 0) THEN
            CALL ShowSevereError('Priority Plant Control specified without equipment list, for scheme='//TRIM(AlphArray(1)))
            ErrorsFound=.true.
            CYCLE
        END IF
        ALLOCATE (LoadBasedScheme(Count)%EquipOperationListName(LoadBasedScheme(Count)%NumLoadRangeLists))
        ALLOCATE (LoadBasedScheme(Count)%LoadRangeUpperLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
        ALLOCATE (LoadBasedScheme(Count)%LoadRangeLowerLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
        DO ListNum = 1, LoadBasedScheme(Count)%NumLoadRangeLists
            LoadBasedScheme(Count)%LoadRangeLowerLimit(ListNum) = ABS(NumArray(ListNum*2-1))
            LoadBasedScheme(Count)%LoadRangeUpperLimit(ListNum) = ABS(NumArray(ListNum*2))
            LoadBasedScheme(Count)%EquipOperationListName(ListNum) = AlphArray(ListNum+1)
        END DO
        LoadBasedScheme(Count)%OpType=CoolingOp
        LoadBasedScheme(Count)%HeaterCool='COOLING'
    END DO

    DO SchemeNum = 1, HLRBO
        CALL GetObjectItem('HEATING LOAD RANGE BASED OPERATION',SchemeNum, &
                          AlphArray,NumAlphas,NumArray,NumNums,IOSTAT)
        IsNotOK=.false.
        IsBlank=.false.
        CALL VerifyName(AlphArray(1),LoadBasedScheme%Name,Count,IsNotOK,IsBlank,'LOAD RANGE BASED OPERATION
Name')
        IF (IsNotOK) THEN
            ErrorsFound=.true.
            IF (IsBlank) AlphArray(1)='xxxxx'
            ENDIF
            Count=Count+1
            LoadBasedScheme(Count)%Name      = AlphArray(1)

            LoadBasedScheme(Count)%NumLoadRangeLists = NumAlphas - 1
            IF (LoadBasedScheme(Count)%NumLoadRangeLists <= 0) THEN
                CALL ShowSevereError('Priority Plant Control specified without equipment list, for scheme='//TRIM(AlphArray(1)))
                ErrorsFound=.true.
                CYCLE
            END IF
            ALLOCATE (LoadBasedScheme(Count)%EquipOperationListName(LoadBasedScheme(Count)%NumLoadRangeLists))
            ALLOCATE (LoadBasedScheme(Count)%LoadRangeUpperLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
            ALLOCATE (LoadBasedScheme(Count)%LoadRangeLowerLimit(LoadBasedScheme(Count)%NumLoadRangeLists))
            DO ListNum = 1, LoadBasedScheme(Count)%NumLoadRangeLists
                LoadBasedScheme(Count)%LoadRangeLowerLimit(ListNum) = ABS(NumArray(ListNum*2-1))
                LoadBasedScheme(Count)%LoadRangeUpperLimit(ListNum) = ABS(NumArray(ListNum*2))
                LoadBasedScheme(Count)%EquipOperationListName(ListNum) = AlphArray(ListNum+1)
            END DO
            LoadBasedScheme(Count)%OpType=HeatingOp
            LoadBasedScheme(Count)%HeaterCool='HEATING'
        END DO

        NumEquipOperationLists = GetNumObjectsFound('LOAD RANGE EQUIPMENT LIST')
        MaxNumMachines_EquipOpLists=0
        IF (NumEquipOperationLists .gt. 0) THEN
            ALLOCATE (EquipOperationList(NumEquipOperationLists))

            DO Num = 1, NumEquipOperationLists
                CALL GetObjectItem('LOAD RANGE EQUIPMENT LIST',Num, &
                                  AlphArray,NumAlphas,NumArray,NumNums,IOSTAT)
                IsNotOK=.false.
                IsBlank=.false.

```

```

    CALL VerifyName(AlphArray(1),EquipOperationList%Name,Num-1,IsNotOK,IsBlank,'LOAD RANGE EQUIPMENT LIST
Name')
    IF ( IsNotOK) THEN
        ErrorsFound=.true.
        IF ( IsBlank) AlphArray(1)='xxxxx'
    ENDIF
    EquipOperationList(Num)%Name      = AlphArray(1)
    EquipOperationList(Num)%NumMachines = (NumAlphas - 1)/2
    IF (EquipOperationList(Num)%NumMachines.le.0) THEN
        CALL ShowSevereError('Equipment List specified without any machines, for equipment list='//TRIM(AlphArray(1)))
        ErrorsFound=.true.
        CYCLE
    END IF
    MaxNumMachines_EquipOpLists=MAX(MaxNumMachines_EquipOpLists,EquipOperationList(Num)%NumMachines)
    ALLOCATE (EquipOperationList(Num)%MachineName(EquipOperationList(Num)%NumMachines))
    ALLOCATE (EquipOperationList(Num)%MachineType(EquipOperationList(Num)%NumMachines))
    DO MachineNum = 1, EquipOperationList(Num)%NumMachines
        EquipOperationList(Num)%MachineType(MachineNum) = AlphArray(MachineNum*2)
        EquipOperationList(Num)%MachineName(MachineNum) = AlphArray(MachineNum*2+1)
    END DO !MachineList
    END DO !EquipOperationList
    END IF

    ! Now can validate Equipment List names (referenced in Operation Schemes)
    DO SchemeNum=1,NumLoadRangeSchemes
        DO ListNum = 1, LoadBasedScheme(SchemeNum)%NumLoadRangeLists
            EquipListNum=FindItemInList(LoadBasedScheme(SchemeNum)%EquipOperationListName(ListNum), &
                                         EquipOperationList%Name,NumEquipOperationLists)
            IF (EquipListNum == 0) THEN
                CALL ShowSevereError('Invalid Equipment Operation List Name='// &
                                     TRIM(LoadBasedScheme(SchemeNum)%EquipOperationListName(ListNum))// &
                                     ', in Load Based Operation Scheme='//TRIM(LoadBasedScheme(SchemeNum)%Name))
                ErrorsFound=.true.
            ENDIF
        ENDDO
    ENDDO
    DO SchemeNum=1,NumUncontrolledSchemes
        DO ListNum = 1, UncontrolledLoopScheme(SchemeNum)%NumLoadRangeLists
            EquipListNum=FindItemInList(UncontrolledLoopScheme(SchemeNum)%EquipOperationListName(ListNum), &
                                         EquipOperationList%Name,NumEquipOperationLists)
            IF (EquipListNum == 0) THEN
                CALL ShowSevereError('Invalid Equipment Operation List Name='// &
                                     TRIM(UncontrolledLoopScheme(SchemeNum)%EquipOperationListName(ListNum))// &
                                     ', in Uncontrolled Loop Operation Scheme='//TRIM(UncontrolledLoopScheme(SchemeNum)%Name))
                ErrorsFound=.true.
            ENDIF
        ENDDO
    ENDDO

    ! Validate that component names/types in each list correspond to a valid component in input file
    IF (ErrorsFound) THEN
        CALL ShowSevereError('Errors found in getting Plant Operation Schemes')
    ENDIF

    RETURN

END SUBROUTINE GetOperationSchemeInput

SUBROUTINE VerifyEquipLists

    ! SUBROUTINE INFORMATION:
    !   AUTHOR      Linda Lawrie
    !   DATE WRITTEN March 2002
    !   MODIFIED    na
    !   RE-ENGINEERED na

```

```

! PURPOSE OF THIS SUBROUTINE:
! This subroutine is called after all operation inputs have been gotten and
! verifies that equipment on the lists match up.

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE ! Enforce explicit typing of all variables in this routine

! SUBROUTINE ARGUMENT DEFINITIONS:
! na

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
LOGICAL :: ErrorsFound=.false.
INTEGER :: Num
INTEGER :: MachineNum
INTEGER :: LoopNum
INTEGER :: BranchNum
INTEGER :: CompNum
CHARACTER(len=MaxNameLength) :: EquipName ! intermediate variable for equipment name in equipment operation list
CHARACTER(len=MaxNameLength) :: EquipType ! intermediate variable for equipment type in equipment operation list
LOGICAL :: FoundMatch ! Used to denote whether or not an equipment name/type found in an equipment operation list
                           ! has been defined by the user as a valid piece of equipment elsewhere in the input

DO Num = 1, NumEquipOperationLists
  DO MachineNum = 1, EquipOperationList(Num)%NumMachines

    FoundMatch = .FALSE.
    EquipName = EquipOperationList(Num)%MachineName(MachineNum)
    EquipType = EquipOperationList(Num)%MachineType(MachineNum)

    ! Must check both Plant and Condenser supply sides
    DO LoopNum = 1, NumPlantLoops
      DO BranchNum = 1, PlantSupplySide(LoopNum)%TotalBranches
        DO CompNum = 1, PlantSupplySide(LoopNum)%Branch(BranchNum)%TotalComponents
          IF ( (EquipType == PlantSupplySide(LoopNum)%Branch(BranchNum)%Comp(CompNum)%TypeOf) .AND. &
               (EquipName == PlantSupplySide(LoopNum)%Branch(BranchNum)%Comp(CompNum)%Name) ) FoundMatch =
        .TRUE.
          IF (FoundMatch) EXIT ! Component DO loop
        END DO
        IF (FoundMatch) EXIT ! Branch DO loop
      END DO
      IF (FoundMatch) EXIT ! Plant Loop DO loop
    END DO

    DO LoopNum = 1, NumCondLoops
      DO BranchNum = 1, CondSupplySide(LoopNum)%TotalBranches
        DO CompNum = 1, CondSupplySide(LoopNum)%Branch(BranchNum)%TotalComponents
          IF ( (EquipType == CondSupplySide(LoopNum)%Branch(BranchNum)%Comp(CompNum)%TypeOf) .AND. &

```

```

        (EquipName == CondSupplySide(LoopNum)%Branch(BranchNum)%Comp(CompNum)%Name) ) FoundMatch =
.TRUE.
    IF (FoundMatch) EXIT ! Component DO loop
    END DO
    IF (FoundMatch) EXIT ! Branch DO loop
    END DO
    IF (FoundMatch) EXIT ! Condenser Loop DO loop
    END DO

    IF (.NOT. FoundMatch) THEN ! No match was found so we need to produce an error message
        ErrorsFound = .TRUE.
        CALL ShowSevereError('Could not find equipment of name '//TRIM(EquipName)//' and type '//TRIM(EquipType))
        CALL ShowContinueError('This device shows up in an equipment operation list but is not defined elsewhere in the input')
    END IF

    END DO
    END DO

    IF (ErrorsFound) THEN
        CALL ShowFatalError('Could not verify equipment (see previous error messages), equipment was not on //' &
                           'Plant Loop or Condenser Loop')
    ENDIF

    RETURN

END SUBROUTINE VerifyEquipLists

! End of GetInput subroutines for the Module
!*****



! Beginning Initialization Section of the Plant Loop Module
!*****



! End Initialization Section of the Plant Loop Module
!*****



! Beginning of Plant Loop Module Utility Subroutines
!*****



SUBROUTINE
FindLoadRangeAndEquipLists(LoopDemand,OperationSchemeName,EquipListNum,LoadBasedSchemeNum,LoadRangeListNum,
ListERROR)
    ! SUBROUTINE INFORMATION:
    !   AUTHOR      Dan Fisher
    !   DATE WRITTEN July 1998
    !   MODIFIED    na
    !   RE-ENGINEERED na

    ! PURPOSE OF THIS SUBROUTINE:
    ! This subroutine sets a logical flag
    ! for each piece of equipment in an operation scheme
    ! telling the simulation whether or not that piece of
    ! equipment is available ('ON') for that iteration.
    ! The equipment is available if it is specified on
    ! the list of equipment for the specified load range.

    ! METHODOLOGY EMPLOYED:
    ! na

    ! REFERENCES:
    ! na

    ! USE STATEMENTS:
USE InputProcessor, ONLY: FindItemInList

```

```

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(OUT) :: EquipListNum

!Changed to inout so that previous run will be saved and we can continue through current run
INTEGER, INTENT(INOUT) :: LoadRangeListNum
REAL , INTENT(IN) :: LoopDemand
INTEGER, INTENT(OUT) :: LoadBasedSchemeNum
LOGICAL, INTENT(INOUT) :: ListERROR
CHARACTER(len=*), INTENT(IN) :: OperationSchemeName

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER,SAVE :: ErrCount=0
LOGICAL :: HeatingCoolingFlag=.True.
CHARACTER(len=20):: CharErrOut
CHARACTER(len=25):: ChrLoopDemand

EquipListNum = 0
ListERROR = .FALSE.
HeatingCoolingFlag = .True.

! If New load range, select available equipment
LoadBasedSchemeNum = FindItemInList(OperationSchemeName, LoadBasedScheme%Name, NumLoadRangeSchemes)
IF (LoadBasedSchemeNum == 0) THEN
  CALL ShowFatalError('Operation Scheme not found='//TRIM(OperationSchemeName))
ENDIF
!Find the appropriate Load Range operation equipment list
DO LoadRangeListNum = 1, LoadBasedScheme(LoadBasedSchemeNum)%NumLoadRangeLists

  !cycle through load ranges until the loop demand is bounded by the range
  IF (LoopDemand < 0.0) THEN !cooling
    IF (LoadBasedScheme(LoadBasedSchemeNum)%OpType == HeatingOp) Then
      HeatingCoolingFlag = .False.
      CYCLE
    End If
    IF (ABS(LoopDemand) < ABS(LoadBasedScheme(LoadBasedSchemeNum)%LoadRangeLowerLimit(LoadRangeListNum)))
  CYCLE
    IF (ABS(LoopDemand) >= ABS(LoadBasedScheme(LoadBasedSchemeNum)%LoadRangeUpperLimit(LoadRangeListNum)))
  CYCLE
  ELSE IF (LoopDemand > 0.0) THEN !Heating
    IF (LoadBasedScheme(LoadBasedSchemeNum)%OpType == CoolingOp) Then
      HeatingCoolingFlag = .False.
      CYCLE
    End If
    IF (LoopDemand < LoadBasedScheme(LoadBasedSchemeNum)%LoadRangeLowerLimit(LoadRangeListNum)) CYCLE
    IF (LoopDemand >= LoadBasedScheme(LoadBasedSchemeNum)%LoadRangeUpperLimit(LoadRangeListNum)) CYCLE
  Else If(LoopDemand == 0.0) THEN
    IF (LoopDemand < LoadBasedScheme(LoadBasedSchemeNum)%LoadRangeLowerLimit(LoadRangeListNum)) CYCLE
  END IF

  !Once the load range has been identified, Search through Equipment
  !Operation Lists until a match with the
  !selected Load Range Operation Scheme is obtained.
  EquipListNum=FindItemInList(LoadBasedScheme(LoadBasedSchemeNum)%EquipOperationListName(LoadRangeListNum),
&

```

```

        EquipOperationList%Name,NumEquipOperationLists)
IF (EquipListNum /= 0) EXIT
END DO !Load Ranges

IF ((EquipListNum == 0) .and. (HeatingCoolingFlag)) THEN
    !No lists were found

    !Since no list was found, assign the largest list, just so demand can be seen,
    ! warning printed as well

    LoadRangeListNum = 0
    EquipListNum     = 0

    ErrCount=ErrCount+1
    IF (ErrCount < 15) THEN
        WRITE(ChrLoopDemand,*) LoopDemand
        ChrLoopDemand=ADJUSTL(ChrLoopDemand)
        IF (INDEX(ChrLoopDemand,'.') > 0) ChrLoopDemand=ChrLoopDemand(1:INDEX(ChrLoopDemand,'.')+3)
        CALL ShowWarningError ('No equipment found in FindLoadRangeAndEquipLists, LoopDemand='//TRIM(ChrLoopDemand)//

&
        ' Operation Scheme='//TRIM(OperationSchemeName))
    ELSE
        IF (MOD(ErrCount,100) == 0) THEN
            WRITE(CharErrOut,*) ErrCount
            CharErrOut=ADJUSTL(CharErrOut)
            CALL ShowWarningError ('No equipment found in FindLoadRangeAndEquipLists continues...'//CharErrOut)
        ENDIF
    ENDIF

    ListERROR = .TRUE.
    ENDIF

    RETURN
END SUBROUTINE FindLoadRangeAndEquipLists

*****
SUBROUTINE TurnOnCondLoopPipes(LoopNum)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE: This subroutine sets a logical flag
! for the condenser loop circulation pump to TRUE

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN) :: LoopNum

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS

```

```

! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER :: MachineOnLoopNum
INTEGER :: Num

DO Num = 1, CondSupplySide(LoopNum)%TotalBranches
DO MachineOnLoopNum = 1, CondSupplySide(LoopNum)%Branch(Num)%TotalComponents
  IF(CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnLoopNum)%TypeOf == 'PIPE')THEN
    CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnLoopNum)%ON = .TRUE.
  END IF
END DO
END DO

RETURN
END SUBROUTINE TurnOnCondLoopPipes
*****



SUBROUTINE TurnOnPlantLoopPipes(LoopNum)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE: This subroutine sets a logical flag
! for the loop circulation pump to TRUE.

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN) :: LoopNum

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER :: MachineOnLoopNum
INTEGER :: Num

DO Num = 1, PlantSupplySide(LoopNum)%TotalBranches
DO MachineOnLoopNum = 1, PlantSupplySide(LoopNum)%Branch(Num)%TotalComponents
  IF(PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnLoopNum)%TypeOf == 'PIPE')THEN
    PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnLoopNum)%ON = .TRUE.
  END IF
END DO
END DO

```

```

RETURN
END SUBROUTINE TurnOnPlantLoopPipes

*****
SUBROUTINE TurnOnAvailableCondEquip(LoopNum,EquipListNum)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE:
! This subroutine sets a logical flag
! for each piece of equipment in an operation scheme
! telling the simulation whether or not that piece of
! equipment is available ('ON') for that iteration.
! The equipment is available if it is specified on
! the list of equipment for the specified load range.

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN)      :: EquipListNum
INTEGER, INTENT(IN)      :: LoopNum

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER          :: AvailableMachineNum
INTEGER          :: Num
INTEGER          :: MachineOnBranch
CHARACTER(len=MaxNameLength) :: AvailableMachineName
CHARACTER(len=MaxNameLength) :: AvailableMachineType
CHARACTER(len=MaxNameLength) :: MachineOnLoopName
CHARACTER(len=MaxNameLength) :: MachineOnLoopType
LOGICAL          :: AvailMachineFound

DO AvailableMachineNum = 1, EquipOperationList(EquipListNum)%NumMachines
  AvailableMachineName = EquipOperationList(EquipListNum)%MachineName(AvailableMachineNum)
  AvailableMachineType = EquipOperationList(EquipListNum)%MachineType(AvailableMachineNum)
  AvailMachineFound = .FALSE.
  DO Num =1, CondSupplySide(LoopNum)%TotalBranches
    DO MachineOnBranch = 1, CondSupplySide(LoopNum)%Branch(Num)%TotalComponents
      MachineOnLoopName = CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%Name
      MachineOnLoopType = CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%TypeOf
      IF (AvailableMachineName == MachineOnLoopName .AND. &
          AvailableMachineType == MachineOnLoopType) THEN
        CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%ON = .TRUE.
      END IF
    END DO
  END DO
END DO

```

```

AvailMachineFound = .TRUE.
EXIT ! DO loop through machines
END IF
END DO ! machines on branch
IF (AvailMachineFound) EXIT ! DO loop through branches, go to next available machine
END DO
END DO !Available Machines

RETURN
END SUBROUTINE TurnOnAvailableCondEquip

SUBROUTINE DistributePlantLoad(LoopNum,EquipListNum,LoopDemand)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE: This subroutine distributes the load
! to plant equipment according to one of two distribution schemes:
!   OPTIMAL   = 1
!   SEQUENTIAL = 2

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

```

IMPLICIT NONE

```

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN) :: EquipListNum
INTEGER, INTENT(IN) :: LoopNum
REAL, INTENT(IN) :: LoopDemand

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
REAL :: RemLoopDemand
REAL :: ChangeInLoad
REAL :: DivideLoad
REAL :: NewLoad
INTEGER :: LoadFlag
INTEGER :: Num
INTEGER :: BNum
INTEGER :: MNum
INTEGER :: MachineOnBranch
INTEGER :: AvailableMachineNum
INTEGER :: MachineCounter
CHARACTER(len=MaxNameLength) :: AvailableMachineName
CHARACTER(len=MaxNameLength) :: AvailableMachineType
CHARACTER(len=MaxNameLength) :: MachineOnLoopName
CHARACTER(len=MaxNameLength) :: MachineOnLoopType

```

```

INTEGER, SAVE, ALLOCATABLE, DIMENSION(:) :: BranchIndex
INTEGER, SAVE, ALLOCATABLE, DIMENSION(:) :: MachineIndex
LOGICAL, SAVE :: FirstTime=.true.

IF (FirstTime) THEN
    ! reallocate the index arrays
    IF(ALLOCATED(BranchIndex))DEALLOCATE(BranchIndex)
    IF(ALLOCATED(MachineIndex))DEALLOCATE(MachineIndex)
    ALLOCATE(BranchIndex(MaxNumMachines_EquipOpLists))
    ALLOCATE(MachineIndex(MaxNumMachines_EquipOpLists))
    FirstTime=.false.
ENDIF

! load local variables
RemLoopDemand = ABS(LoopDemand)
LoadFlag = plantsupplyside(LoopNum)%LOADDISTRIBUTION
MachineCounter = 0
BranchIndex=0
MachineIndex=0

! search equiplist for operating machines and save their indices
DO AvailableMachineNum = 1, EquipOperationList(EquipListNum)%NumMachines
    AvailableMachineName = EquipOperationList(EquipListNum)%MachineName(AvailableMachineNum)
    AvailableMachineType = EquipOperationList(EquipListNum)%MachineType(AvailableMachineNum)
    DO Num =1, PlantSupplySide(LoopNum)%TotalBranches
        DO MachineOnBranch = 1, PlantSupplySide(LoopNum)%Branch(Num)%TotalComponents
            MachineOnLoopName = PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%Name
            MachineOnLoopType = PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%TypeOf

IF (AvailableMachineName == MachineOnLoopName .AND.      &
    AvailableMachineType == MachineOnLoopType) THEN

    PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%ON = .TRUE.
    PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%MyLoad = 0.0
    MachineCounter =MachineCounter+1
    BranchIndex(MachineCounter)=Num
    MachineIndex(MachineCounter)=MachineOnBranch
    EXIT
END IF
END DO ! machines on branch
END DO
END DO !Available Machines

!OPTIMAL DISTRIBUTION SCHEME
SELECT CASE (LoadFlag)
CASE (1) ! LoadFlag=1 indicates "optimal" load distribution

    !step 1: load all machines to optimal PLR
    DO Num =1, MachineCounter
        BNum = BranchIndex(Num)
        MNum = MachineIndex(Num)
        ! For the case where there is no Loop Demand but the pumps are running
        IF (RemLoopDemand == 0.0)EXIT

        !Then we continue with the normal case
        ChangeInLoad= MIN(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Optload,RemLoopDemand)
        ChangeInLoad= MAX(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Minload,ChangeInLoad)
        PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = ChangeInLoad
        RemLoopDemand = RemLoopDemand - PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload
        IF (RemLoopDemand <= 0.0)EXIT
    END DO

    !step 2: Evenly distribute remaining loop demand
    IF (RemLoopDemand > 0.0)THEN
        If(MachineCounter > 0) DivideLoad = RemLoopDemand/MachineCounter
        DO Num =1, MachineCounter

```

```

BNum = BranchIndex(Num)
MNum = MachineIndex(Num)

NewLoad = PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload

NewLoad = MIN(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Maxload, &
             NewLoad + DivideLoad)

ChangeInLoad = NewLoad - PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload
PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = NewLoad

RemLoopDemand= RemLoopDemand - ChangeInLoad
IF (RemLoopDemand <= 0.0)EXIT
END DO
END IF

! step 3: If RemLoopDemand is still greater than zero, look for any machine
IF (RemLoopDemand > 0.0)THEN

DO Num =1, MachineCounter
BNum = BranchIndex(Num)
MNum = MachineIndex(Num)

DivideLoad = PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Maxload &
            - PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload

ChangeInLoad = MIN(RemLoopDemand, DivideLoad)

PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = &
PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload + ChangeInLoad
RemLoopDemand= RemLoopDemand - ChangeInLoad
IF (RemLoopDemand <= 0.0)EXIT
END DO
END IF

! step 4: If RemLoopDemand is less than zero, decrement load on first machine
IF (RemLoopDemand < 0.0)THEN

BNum = BranchIndex(1)
MNum = MachineIndex(1)

ChangeInLoad = PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload - abs(RemLoopDemand)
PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = &
MAX(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Minload,ChangeInLoad)
END IF

!SEQUENTIAL DISTRIBUTION SCHEME
CASE (2)! LoadFlag=2 indicates "sequential" load distribution

! step 1: If RemLoopDemand is greater than zero, load machines in list order
IF (RemLoopDemand > 0.0)THEN

DO Num =1, MachineCounter
BNum = BranchIndex(Num)
MNum = MachineIndex(Num)

ChangeInLoad = MIN(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Maxload, &
                  RemLoopDemand)
ChangeInLoad = MAX(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Minload, &
                  ChangeInLoad)

PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = ChangeInLoad

RemLoopDemand= RemLoopDemand - ChangeInLoad
IF (RemLoopDemand <= 0.0)EXIT
END DO

```

```

END IF

! step 1: If RemLoopDemand is less than zero, decrement load on last machine
IF (RemLoopDemand < 0.0)THEN

    BNum = BranchIndex(1)
    MNum = MachineIndex(1)

    ChangeInLoad = PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload - RemLoopDemand
    PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Myload = &
    MAX(PlantSupplySide(LoopNum)%Branch(BNum)%Comp(MNum)%Minload,ChangeInLoad)
END IF

End Select

RETURN
END SUBROUTINE DistributePlantLoad

SUBROUTINE TurnOffPlantEquipment(LoopNum)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE:

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN)      :: LoopNum

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER                  :: MachineOnBranch
INTEGER                  :: Num

!FLOW

DO Num =1, PlantSupplySide(LoopNum)%TotalBranches
DO MachineOnBranch = 1, PlantSupplySide(LoopNum)%Branch(Num)%TotalComponents
IF (PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%TypeOf(1:4) /= 'PUMP') THEN
    PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%ON = .FALSE.
    PlantSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%Myload = 0.0
End If
END DO ! machines on branch
END DO

```

```

RETURN
END SUBROUTINE TurnOffPlantEquipment

SUBROUTINE TurnOffCondEquipment(LoopNum)
! SUBROUTINE INFORMATION:
!   AUTHOR      Dan Fisher
!   DATE WRITTEN July 1998
!   MODIFIED    na
!   RE-ENGINEERED na

! PURPOSE OF THIS SUBROUTINE:

! METHODOLOGY EMPLOYED:
! na

! REFERENCES:
! na

! USE STATEMENTS:
! na

IMPLICIT NONE

! SUBROUTINE ARGUMENT DEFINITIONS:
INTEGER, INTENT(IN)      :: LoopNum

! SUBROUTINE PARAMETER DEFINITIONS:
! na

! INTERFACE BLOCK SPECIFICATIONS
! na

! DERIVED TYPE DEFINITIONS
! na

! SUBROUTINE LOCAL VARIABLE DECLARATIONS:
INTEGER                  :: MachineOnBranch
INTEGER                  :: Num

!FLOW

DO Num =1, CondSupplySide(LoopNum)%TotalBranches
DO MachineOnBranch = 1, CondSupplySide(LoopNum)%Branch(Num)%TotalComponents
IF (CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%TypeOf(1:4) /= 'PUMP') THEN
  CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%ON = .FALSE.
  CondSupplySide(LoopNum)%Branch(Num)%Comp(MachineOnBranch)%Myload = 0.0
End If
END DO ! machines on branch
END DO

RETURN
END SUBROUTINE TurnOffCondEquipment

END MODULE PlantCondLoopOperation

```

## **ANNEXE 2**

**Sommaire des résultats avec un pas de temps de calculs de 10 minutes**

Tableau VIII

Modèle #1 – Option 1 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				
			Redevance	Frais de puissance	Frais d'énergie		Total
					1 <sup>ère</sup> tranche	2 <sup>ième</sup> tranche	
Janv.	11731	38	11,67 \$	67,40 \$	866,97 \$	1,15 \$	947,19 \$
Fév.	10308	38	11,67 \$	67,40 \$	763,82 \$	- \$	842,89 \$
Mars	11350	38	11,67 \$	67,40 \$	841,03 \$	- \$	920,10 \$
Avril	11670	39	11,67 \$	67,40 \$	864,72 \$	- \$	943,78 \$
Mai	13000	56	11,67 \$	212,18 \$	866,97 \$	48,61 \$	1 139,42 \$
Juin	14044	58	11,67 \$	247,50 \$	866,97 \$	87,66 \$	1 213,80 \$
Juil.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$
Août	15690	63	11,67 \$	308,53 \$	866,97 \$	149,23 \$	1 336,41 \$
Sept.	12374	59	11,67 \$	253,02 \$	866,97 \$	25,21 \$	1 156,86 \$
Oct.	12742	54	11,67 \$	190,50 \$	866,97 \$	38,97 \$	1 108,11 \$
Nov.	11279	42	11,67 \$	67,40 \$	835,80 \$	- \$	914,87 \$
Déc.	10997	38	11,67 \$	67,40 \$	814,91 \$	- \$	893,98 \$

Grand Total : 12 861,96 \$

Tableau IX

Modèle #1 – Option 1 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif BT				
			Redevance	Frais de puissance	Frais d'énergie		Total
					Hors pointe	Pointe	
Janv.	6476	55	33,15 \$	4,73 \$	205,03 \$	31,69 \$	274,60 \$
Fév.	5232	63	33,15 \$	4,73 \$	165,20 \$	31,91 \$	234,99 \$
Mars	5936	59	33,15 \$	4,73 \$	189,94 \$	- \$	227,82 \$
Avril	3608	47	33,15 \$	4,73 \$	115,45 \$	- \$	153,33 \$
Mai	3344	27	33,15 \$	4,73 \$	107,00 \$	- \$	144,88 \$
Juin	2823	17	33,15 \$	4,73 \$	90,34 \$	- \$	128,22 \$
JUIL.	2702	13	33,15 \$	4,73 \$	86,46 \$	- \$	124,34 \$
Août	2865	14	33,15 \$	4,73 \$	91,67 \$	- \$	129,55 \$
Sept.	3121	16	33,15 \$	4,73 \$	99,88 \$	- \$	137,75 \$
Oct.	3876	45	33,15 \$	4,73 \$	124,02 \$	- \$	161,89 \$
Nov.	5286	49	33,15 \$	4,73 \$	169,15 \$	- \$	207,03 \$
Déc.	7681	61	33,15 \$	4,73 \$	243,53 \$	2,26 \$	283,67 \$

Grand Total : 2 208,08 \$

Tableau X

## Modèle #1 – Option 1 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes

<b>Mois</b>	<b>Consommation [m<sup>3</sup>]</b>	<b>Coûts</b>
Janvier	1534	552,39 \$
Février	1397	502,91 \$
Mars	193	69,53 \$
Avril	0	- \$
Mai	0	- \$
Juin	0	- \$
Juillet	0	- \$
Août	0	- \$
Septembre	0	- \$
Octobre	0	- \$
Novembre	0	- \$
Décembre	941	338,89 \$

**Grand Total : 1 463,71 \$**

Tableau XI

## Modèle #1 – Option 2 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G					Total	
			Redevance	Frais de puissance	Frais d'énergie				
					1ère tranche	2ième tranche			
Janv.	11731	38	11,67 \$	67,40 \$	866,97 \$	1,15 \$	947,19 \$		
Fév.	10308	38	11,67 \$	67,40 \$	763,82 \$	- \$	842,89 \$		
Mars	11350	38	11,67 \$	67,40 \$	841,03 \$	- \$	920,10 \$		
Avril	11670	39	11,67 \$	67,40 \$	864,72 \$	- \$	943,78 \$		
Mai	13000	56	11,67 \$	212,18 \$	866,97 \$	48,61 \$	1 139,42 \$		
Juin	14044	58	11,67 \$	247,50 \$	866,97 \$	87,66 \$	1 213,80 \$		
Juil.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$		
Août	15690	63	11,67 \$	308,53 \$	866,97 \$	149,23 \$	1 336,41 \$		
Sept.	12374	59	11,67 \$	253,02 \$	866,97 \$	25,21 \$	1 156,86 \$		
Oct.	12742	54	11,67 \$	190,50 \$	866,97 \$	38,97 \$	1 108,11 \$		
Nov.	11279	42	11,67 \$	67,40 \$	835,80 \$	- \$	914,87 \$		
Déc.	10997	38	11,67 \$	67,40 \$	814,91 \$	- \$	893,98 \$		

**Grand Total : 12 861,96 \$**

Tableau XII

Modèle #1 – Option 2 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif BT				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					Hors pointe	Pointe		
Janv.	6476	55	33,15 \$	4,73 \$	199,60 \$	61,83 \$	299,31 \$	
Fév.	5232	63	33,15 \$	4,73 \$	160,24 \$	103,30 \$	301,42 \$	
Mars	5936	59	33,15 \$	4,73 \$	186,44 \$	50,42 \$	274,73 \$	
Avril	3608	47	33,15 \$	4,73 \$	115,45 \$	- \$	153,33 \$	
Mai	3344	27	33,15 \$	4,73 \$	107,00 \$	- \$	144,88 \$	
Juin	2823	17	33,15 \$	4,73 \$	90,34 \$	- \$	128,22 \$	
Juil.	2702	13	33,15 \$	4,73 \$	86,46 \$	- \$	124,34 \$	
Août	2865	14	33,15 \$	4,73 \$	91,67 \$	- \$	129,55 \$	
Sept.	3121	16	33,15 \$	4,73 \$	99,88 \$	- \$	137,75 \$	
Oct.	3876	45	33,15 \$	4,73 \$	124,02 \$	- \$	161,89 \$	
Nov.	5286	49	33,15 \$	4,73 \$	169,15 \$	- \$	207,03 \$	
Déc.	7681	61	33,15 \$	4,73 \$	241,51 \$	61,59 \$	340,98 \$	

**Grand Total : 2403,44 \$**

Tableau XIII

Modèle #1 – Option 2 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes

Mois	Consommation [m <sup>3</sup> ]	Coûts
Janvier	1534	552,39 \$
Février	1397	502,91 \$
Mars	193	69,53 \$
Avril	0	- \$
Mai	0	- \$
Juin	0	- \$
JUILLET	0	- \$
Août	0	- \$
Septembre	0	- \$
Octobre	0	- \$
Novembre	0	- \$
Décembre	941	338,89 \$

**Grand Total : 1 463,71 \$**

Tableau XIV

Modèle #2 : Coûts d'électricité avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				
			Redevance	Frais de puissance	Frais d'énergie		Total
					1ère tranche	2ième tranche	
Janv.	23474	107	11,67 \$	917,18 \$	866,97 \$	440,33 \$	2 236,15 \$
Fév.	20013	105	11,67 \$	878,90 \$	866,97 \$	310,91 \$	2 068,45 \$
Mars	17836	92	11,67 \$	713,11 \$	866,97 \$	229,50 \$	1 821,25 \$
Avril	15277	82	11,67 \$	574,48 \$	866,97 \$	133,80 \$	1 586,91 \$
Mai	16343	65	11,67 \$	405,91 \$	866,97 \$	173,66 \$	1 458,21 \$
Juin	16867	67	11,67 \$	405,91 \$	866,97 \$	193,25 \$	1 477,80 \$
Juil.	18934	69	11,67 \$	405,91 \$	866,97 \$	270,56 \$	1 555,10 \$
Août	18555	65	11,67 \$	405,91 \$	866,97 \$	256,37 \$	1 540,92 \$
Sept.	15495	61	11,67 \$	405,91 \$	866,97 \$	141,94 \$	1 426,48 \$
Oct.	16617	81	11,67 \$	552,19 \$	866,97 \$	183,91 \$	1 614,75 \$
Nov.	16565	82	11,67 \$	572,29 \$	866,97 \$	181,97 \$	1 632,89 \$
Déc.	21640	96	11,67 \$	758,20 \$	866,97 \$	371,76 \$	2 008,59 \$

Grand Total : 20 427,51 \$

Tableau XV

Modèle #3 : Coûts d'électricité avec un pas de temps de calculs de 10 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				
			Redevance	Frais de puissance	Frais d'énergie		Total
					1ère tranche	2ième tranche	
Janv.	11731	38	11,67 \$	- \$	866,97 \$	1,15 \$	879,79 \$
Fév.	10308	38	11,67 \$	- \$	763,82 \$	- \$	775,49 \$
Mars	11350	38	11,67 \$	- \$	841,03 \$	- \$	852,70 \$
Avril	11670	39	11,67 \$	- \$	864,72 \$	- \$	876,39 \$
Mai	13000	56	11,67 \$	212,18 \$	866,97 \$	48,61 \$	1 139,42 \$
Juin	14044	58	11,67 \$	247,50 \$	866,97 \$	87,66 \$	1 213,80 \$
JUIL.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$
Août	15690	63	11,67 \$	308,53 \$	866,97 \$	149,23 \$	1 336,41 \$
Sept.	12374	59	11,67 \$	253,02 \$	866,97 \$	25,21 \$	1 156,86 \$
Oct.	12742	54	11,67 \$	190,50 \$	866,97 \$	38,97 \$	1 108,11 \$
Nov.	11279	42	11,67 \$	24,45 \$	835,80 \$	- \$	871,93 \$
Déc.	10997	38	11,67 \$	- \$	814,91 \$	- \$	826,58 \$

Grand Total : 12 482,03 \$

Tableau XVI

Modèle #3 : Coûts du gaz naturel avec un pas de temps de calculs de 10 minutes

<b>Mois</b>	<b>Consommation [m<sup>3</sup>]</b>	<b>Coûts</b>
Janvier	3912	1 408,32 \$
Février	3340	1 202,32 \$
Mars	2481	893,15 \$
Avril	1481	533,30 \$
Mai	1404	505,58 \$
Juin	1187	427,18 \$
JUILLET	1136	408,84 \$
Août	1204	433,47 \$
Septembre	1312	472,28 \$
Octobre	1595	574,04 \$
Novembre	2063	742,61 \$
Décembre	3713	1 336,81 \$

**Grand Total : 8 937,88 \$**

### **ANNEXE 3**

**Sommaire des résultats avec un pas de temps de calculs de 60 minutes**

Tableau XVII

Modèle #1 – Option 1 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					1 <sup>ère</sup> tranche	2 <sup>ième</sup> tranche		
Janv.	11731	38	11,67 \$	- \$	866,97 \$	1,15 \$	879,79 \$	
Fév.	10308	38	11,67 \$	- \$	763,82 \$	- \$	775,49 \$	
Mars	11350	38	11,67 \$	- \$	841,03 \$	- \$	852,70 \$	
Avril	11670	39	11,67 \$	- \$	864,72 \$	- \$	876,39 \$	
Mai	13000	55	11,67 \$	203,01 \$	866,97 \$	48,61 \$	1 130,26 \$	
Juin	14044	58	11,67 \$	239,32 \$	866,97 \$	87,66 \$	1 205,62 \$	
Juil.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$	
Août	15690	62	11,67 \$	297,86 \$	866,97 \$	149,23 \$	1 325,73 \$	
Sept.	12374	59	11,67 \$	251,64 \$	866,97 \$	25,21 \$	1 155,49 \$	
Oct.	12742	54	11,67 \$	190,39 \$	866,97 \$	38,97 \$	1 108,00 \$	
Nov.	11279	40	11,67 \$	3,40 \$	835,80 \$	- \$	850,87 \$	
Déc.	10997	38	11,67 \$	- \$	814,91 \$	- \$	826,58 \$	

Grand Total : 12 431,46 \$

Tableau XVIII

Modèle #1 – Option 1 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif BT				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					Hors pointe	Pointe		
Janv.	6476	44	33,15 \$	4,73 \$	201,53 \$	77,28 \$	316,69 \$	
Fév.	5232	52	33,15 \$	4,73 \$	160,02 \$	103,68 \$	301,58 \$	
Mars	5936	50	33,15 \$	4,73 \$	189,94 \$	0,95 \$	227,82 \$	
Avril	3608	35	33,15 \$	4,73 \$	115,45 \$	- \$	153,33 \$	
Mai	3344	17	33,15 \$	4,73 \$	107,00 \$	- \$	144,88 \$	
Juin	2823	12	33,15 \$	4,73 \$	90,34 \$	- \$	128,22 \$	
Juil.	2702	12	33,15 \$	4,73 \$	86,46 \$	- \$	124,34 \$	
Août	2865	12	33,15 \$	4,73 \$	91,67 \$	- \$	129,55 \$	
Sept.	3121	12	33,15 \$	4,73 \$	99,88 \$	- \$	137,75 \$	
Oct.	3876	35	33,15 \$	4,73 \$	124,02 \$	- \$	161,89 \$	
Nov.	5286	38	33,15 \$	4,73 \$	169,15 \$	- \$	207,03 \$	
Déc.	7681	53	33,15 \$	4,73 \$	241,27 \$	4,53 \$	283,67 \$	

Grand Total : 2 316,75 \$

## Tableau XIX

## Modèle #1 – Option 1 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes

Mois	Consommation [m <sup>3</sup> ]	Coûts
Janvier	1534	552,39 \$
Février	1397	502,91 \$
Mars	193	69,53 \$
Avril	0	- \$
Mai	0	- \$
Juin	0	- \$
JUILLET	0	- \$
Août	0	- \$
Septembre	0	- \$
Octobre	0	- \$
Novembre	0	- \$
Décembre	941	338,89 \$
<b>Grand Total :</b>		<b>1 463,71 \$</b>

Tableau XX

## Modèle #1 – Option 2 : Coûts d'électricité excluant la chaudière avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					1ère tranche	2ième tranche		
Janv.	11731	38	11,67 \$	- \$	866,97 \$	1,15 \$	879,79 \$	
Fév.	10308	38	11,67 \$	- \$	763,82 \$	- \$	775,49 \$	
Mars	11350	38	11,67 \$	- \$	841,03 \$	- \$	852,70 \$	
Avril	11670	39	11,67 \$	- \$	864,72 \$	- \$	876,39 \$	
Mai	13000	55	11,67 \$	203,01 \$	866,97 \$	48,61 \$	1 130,26 \$	
Juin	14044	58	11,67 \$	239,32 \$	866,97 \$	87,66 \$	1 205,62 \$	
Juil.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$	
Août	15690	62	11,67 \$	297,86 \$	866,97 \$	149,23 \$	1 325,73 \$	
Sept.	12374	59	11,67 \$	251,64 \$	866,97 \$	25,21 \$	1 155,49 \$	
Oct.	12742	54	11,67 \$	190,39 \$	866,97 \$	38,97 \$	1 108,00 \$	
Nov.	11279	40	11,67 \$	3,40 \$	835,80 \$	- \$	850,87 \$	
Déc.	10997	38	11,67 \$	- \$	814,91 \$	- \$	826,58 \$	

**Grand Total : 12 431,46 \$**

Tableau XXI

Modèle #1 – Option 2 : Coûts d'électricité de la chaudière avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif BT				Total
			Redevance	Frais de puissance	Frais d'énergie Hors pointe	Frais d'énergie Pointe	
Janv.	6476	44	33,15 \$	4,73 \$	196,58 \$	121,41 \$	355,87 \$
Fév.	5232	52	33,15 \$	4,73 \$	156,44 \$	157,93 \$	352,24 \$
Mars	5936	50	33,15 \$	4,73 \$	186,21 \$	53,73 \$	277,82 \$
Avril	3608	35	33,15 \$	4,73 \$	115,45 \$	- \$	153,33 \$
Mai	3344	17	33,15 \$	4,73 \$	107,00 \$	- \$	144,88 \$
Juin	2823	12	33,15 \$	4,73 \$	90,34 \$	- \$	128,22 \$
Juil.	2702	12	33,15 \$	4,73 \$	86,46 \$	- \$	124,34 \$
Août	2865	12	33,15 \$	4,73 \$	91,67 \$	- \$	129,55 \$
Sept.	3121	12	33,15 \$	4,73 \$	99,88 \$	- \$	137,75 \$
Oct.	3876	35	33,15 \$	4,73 \$	124,02 \$	- \$	161,89 \$
Nov.	5286	38	33,15 \$	4,73 \$	169,15 \$	- \$	207,03 \$
Déc.	7681	53	33,15 \$	4,73 \$	238,80 \$	100,55 \$	377,23 \$

**Grand Total : 2 550,15 \$**

Tableau XXII

Modèle #1 – Option 2 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes

Mois	Consommation [m³]	Coûts
Janvier	1534	552,39 \$
Février	1397	502,91 \$
Mars	193	69,53 \$
Avril	0	- \$
Mai	0	- \$
Juin	0	- \$
JUILLET	0	- \$
Août	0	- \$
Septembre	0	- \$
Octobre	0	- \$
Novembre	0	- \$
Décembre	941	338,89 \$

**Grand Total : 1 463,71 \$**

Tableau XXIII

Modèle #2 : Coûts d'électricité avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					1ère tranche	2ième tranche		
Janv.	23474	97	11,67 \$	778,30 \$	866,97 \$	440,33 \$	2 097,27 \$	
Fév.	20013	90	11,67 \$	681,64 \$	866,97 \$	310,91 \$	1 871,18 \$	
Mars	17836	76	11,67 \$	488,19 \$	866,97 \$	229,50 \$	1 596,33 \$	
Avril	15277	66	11,67 \$	347,25 \$	866,97 \$	133,80 \$	1 359,69 \$	
Mai	16343	69	11,67 \$	328,67 \$	866,97 \$	173,66 \$	1 380,97 \$	
Juin	16867	64	11,67 \$	354,96 \$	866,97 \$	193,25 \$	1 426,85 \$	
Juil.	18934	66	11,67 \$	396,72 \$	866,97 \$	270,56 \$	1 545,92 \$	
Août	18555	69	11,67 \$	327,28 \$	866,97 \$	256,37 \$	1 462,30 \$	
Sept.	15495	64	11,67 \$	315,64 \$	866,97 \$	141,94 \$	1 336,21 \$	
Oct.	16617	67	11,67 \$	363,80 \$	866,97 \$	183,91 \$	1 426,35 \$	
Nov.	16565	71	11,67 \$	424,53 \$	866,97 \$	181,97 \$	1 485,14 \$	
Déc.	21640	82	11,67 \$	570,15 \$	866,97 \$	371,76 \$	1 820,55 \$	

Grand Total : 18 808,76 \$

Tableau XXIV

Modèle #3 : Coûts d'électricité avec un pas de temps de calculs de 60 minutes

Mois	Consommation [kWh]	Appel de puissance [kW]	Tarif G				Total	
			Redevance	Frais de puissance	Frais d'énergie			
					1ère tranche	2ième tranche		
Janv.	11731	38	11,67 \$	- \$	866,97 \$	1,15 \$	879,79 \$	
Fév.	10308	38	11,67 \$	- \$	763,82 \$	- \$	775,49 \$	
Mars	11350	38	11,67 \$	- \$	841,03 \$	- \$	852,70 \$	
Avril	11670	39	11,67 \$	- \$	864,72 \$	- \$	876,39 \$	
Mai	13000	55	11,67 \$	203,01 \$	866,97 \$	48,61 \$	1 130,26 \$	
Juin	14044	58	11,67 \$	239,32 \$	866,97 \$	87,66 \$	1 205,62 \$	
JUIL.	16232	69	11,67 \$	396,40 \$	866,97 \$	169,50 \$	1 444,54 \$	
Août	15690	62	11,67 \$	297,86 \$	866,97 \$	149,23 \$	1 325,73 \$	
Sept.	12374	59	11,67 \$	251,64 \$	866,97 \$	25,21 \$	1 155,49 \$	
Oct.	12742	54	11,67 \$	190,39 \$	866,97 \$	38,97 \$	1 108,00 \$	
Nov.	11279	40	11,67 \$	3,40 \$	835,80 \$	- \$	850,87 \$	
Déc.	10997	38	11,67 \$	- \$	814,91 \$	- \$	826,58 \$	

Grand Total : 12 431,46 \$

Tableau XXV

Modèle #3 : Coûts du gaz naturel avec un pas de temps de calculs de 60 minutes

Mois	Consommation [m <sup>3</sup> ]	Coûts
Janvier	3912	1 408,32 \$
Février	3340	1 202,32 \$
Mars	2481	893,15 \$
Avril	1481	533,30 \$
Mai	1404	505,58 \$
Juin	1187	427,18 \$
Juillet	1136	408,84 \$
Août	1204	433,47 \$
Septembre	1312	472,28 \$
Octobre	1595	574,04 \$
Novembre	2063	742,61 \$
Décembre	3713	1 336,81 \$
<b>Grand Total :</b> 8 937,88 \$		

## **ANNEXE 4**

### **Publication**

Cet article a été présenté le 12 septembre 2002 dans le cadre de la conférence eSim 2002 qui se tenait à l'Université Concordia, Montréal.

Cet article a également été publié au mois de novembre dans la revue Énergie de l'Association Québécoise pour la Maîtrise de l'Énergie (AQME).

## MODÉLISATION DE SYSTÈMES CVCA À L'AIDE DU LOGICIEL ENERGYPLUS

Bellemare René, Kajl Stanislaw, Roberge Marc-Antoine

École de Technologie Supérieure, 1100, rue Notre-Dame Ouest, Montréal H1C 1K3, Canada.

(514) 396-8517, fax. (514) 396-8530, courriel : [skajl@mec.etsmtl.ca](mailto:skajl@mec.etsmtl.ca)

Dupras Ledoux ingénieurs, 8790, avenue du Parc, Montréal H2N 1Y6, Canada.

(514) 381-9205, fax. (514) 381-1350, courriel : [rbellemare@dupras.com](mailto:rbellemare@dupras.com)

### RÉSUMÉ

L'article démontre les capacités de modélisation d'un système de traitement de l'air d'un bâtiment à l'aide du logiciel EnergyPlus v.1 build23. Pour ce faire, une partie du bâtiment abritant l'École de Technologie Supérieure (ETS) a servi de modèle. Un des onze systèmes VAV et les zones qu'il dessert ont été simulés à l'aide des logiciels EnergyPlus v.1 build23 et DOE-2E. Il s'agit de 54 zones internes formées principalement de salles de cours et de laboratoires informatiques. La première partie de l'article présente la validation dudit système modélisé dans le logiciel EnergyPlus à l'aide d'une base de données enregistrées sur 4 zones internes et sur le système CVCA qui les alimente. Parmi les paramètres enregistrés et comparés, on retrouve la température d'alimentation, le débit d'air soufflé et la température des 4 zones mentionnées précédemment. Les résultats de cette analyse comparative permettent d'affirmer qu'il est possible, à partir d'un relevé détaillé des zones d'un bâtiment, de simuler avec précision l'opération des systèmes CVCA et les zones qu'il dessert.

On démontre également dans cet article l'analyse comparative entre les résultats de simulation obtenus à partir d'un logiciel de référence (DOE-2E) et d'EnergyPlus v.1 build23. On y compare la température ambiante des zones, le débit d'air soufflée dans celles-ci et certains paramètres de fonctionnement du système CVCA. Les résultats de cette deuxième analyse permettent de remarquer de légères différences entre les résultats des deux logiciels.

Finalement, on y commente les particularités et les observations faites lors de la programmation et de l'analyse des résultats. Par exemple, le transfert d'air entre deux zones, l'outil de visualisation des parois introduites dans EnergyPlus, le convertisseur de fichiers météorologiques et le convertisseur de code DOE-2 ont été testés avec succès. Certaines anomalies détectées lors de l'analyse des résultats sont également présentées dans cette section.

### INTRODUCTION

Au cours de la dernière décennie, un nombre important de logiciels de simulation des bâtiments a vu le jour. La

précision de ces logiciels peut varier en fonction de la méthode de calculs et des hypothèses simplificatrices utilisées. En avril 2001, un nouveau logiciel est venu s'ajouter à la liste des logiciels les plus performants, tels DOE-2E et Blast. Contrairement à ses prédecesseurs, ce nouveau logiciel, nommé EnergyPlus version 1.0.0.023, permet de simuler toutes sortes d'arrangements de systèmes mécaniques du bâtiment. Il devenait donc intéressant de tester les capacités de ce dernier en créant un projet d'étude. Ce projet comporte les étapes suivantes :

1. L'analyse comparative du logiciel EnergyPlus.
2. Le développement d'un module de simulation.
3. La validation ainsi que l'intégration du module de simulation dans EnergyPlus.

Dans cet article, seule la première étape de ce projet d'étude est présentée. Après avoir démontré la méthodologie utilisée dans le projet, on présente l'application du logiciel EnergyPlus pour modéliser un système CVCA d'un bâtiment existant. Par la suite, les résultats de deux comparaisons y sont présentés. La première comparaison est réalisée pour valider le modèle. On y compare les résultats obtenus par le logiciel EnergyPlus avec les données enregistrées sur le système CVCA simulé. Dans le but d'éprouver le logiciel davantage, la deuxième comparaison consiste à confronter les résultats obtenus par le logiciel EnergyPlus à ceux obtenus à l'aide du logiciel DOE-2 considéré comme le logiciel de référence. Pour finir, on y présente un résumé des principales remarques et observations rapportées lors de la modélisation.

### MÉTHODOLOGIE

La méthodologie utilisée pour effectuer l'analyse du logiciel EnergyPlus peut être résumée de la façon suivante :

1. Choisir un système CVCA dont le fonctionnement et les caractéristiques mécaniques sont connus et précis;
2. Enregistrer les paramètres d'états et de fonctionnement (température, débit, etc.) de certaines zones sélectionnées et du système CVCA pour une période d'un mois;

3. Modéliser le plus précisément possible à l'aide d'EnergyPlus le système sélectionné et les zones qu'il alimente;
4. Valider les résultats de la simulation avec les données enregistrées;
5. Modéliser le système et les zones qu'il dessert à l'aide du logiciel DOE-2E;
6. Ajuster les deux modèles de simulation de façon à simuler l'opération du système de façon identique dans les deux cas;
7. Comparer les résultats des simulations effectuées à l'aide des deux logiciels;
8. Décrire les éléments positifs et les lacunes du logiciel EnergyPlus rencontrés lors de la modélisation et de l'analyse des résultats.

## MODÉLISATION

Le bâtiment étudié abrite l'École de Technologie Supérieure (ETS). L'édifice est équipé de 11 systèmes de traitement de l'air (CVCA) à volume d'air et température d'alimentation variables. Ils sont alimentés en eau chaude par le biais de la centrale de vapeur de la communauté urbaine de Montréal et en eau froide par le biais d'un groupe refroidisseur centrifuge. Pour limiter la quantité de données à saisir, un (1) système sur 11 a été programmé et simulé soit le système UTA-6. Ce système alimente 54 zones internes du 2<sup>ème</sup> étage de l'édifice. Ces dernières sont principalement composées de salles de cours et de laboratoires informatiques et pratiques.

Lors de la saisie des données dans le logiciel EnergyPlus, le zonage utilisé a été calqué sur le zonage existant du bâtiment. Une zone est définie comme une pièce ou un groupe de pièces desservies par une même boîte à volume d'air variable (VAV). Tel que mentionné ci-dessus, le système CVCA alimentait 54 zones thermiques tandis que le logiciel EnergyPlus permet de connecter un maximum de 50 zones sur un même système CVCA (le nombre maximum de zones est limité par les éléments « Zone Mixer » et « Zone Splitter »). Ainsi, toutes les zones corridors ont été regroupées en une seule zone thermique. Les débits maximum et minimum d'alimentation de chaque zone ont été fixés à partir du rapport d'état des boîtes à volume d'air variable fourni par le gestionnaire du bâtiment. La figure 1 présente le schéma de l'installation du système UTA-6 incluant le nom des différents nœuds modélisés dans EnergyPlus. Des équipements similaires ont également été modélisés à l'aide de DOE-2E.

Les cédules d'occupation des locaux ont été programmées en fonction de l'horaire des activités académiques. Pour les locaux ayant d'autres vocations, un relevé a été fait sur place afin de connaître le

nombre d'occupants, le type d'activités et l'horaire d'occupation. Les cédules d'opération de l'éclairage ont été déterminées à l'aide des cédules d'occupation des locaux et de la cédule de fermeture de l'éclairage de la commande centralisée du bâtiment. La puissance et le type d'éclairage des zones ont, quant à eux, été évalués en fonction des plans électriques de l'édifice. Les puissances d'équipements ont été évaluées de façon sommaire. Toutes les salles de cours sont équipées d'un projecteur vidéo, d'un projecteur d'acétates et d'un ordinateur. Seul l'ordinateur a été considéré, car c'est le seul équipement qui demeure allumé en tout temps, les autres étant utilisés de façon ponctuelle. Pour les autres locaux, un relevé d'équipements a dû être fait. À l'aide des informations données par les occupants, les horaires d'opération des équipements relevés ont pu être établis.

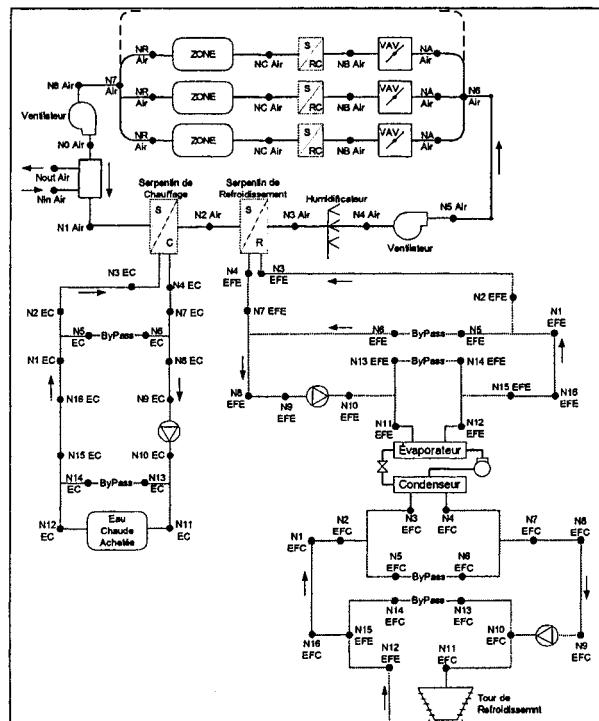


Figure 1. Schéma du système secondaire UTA-6 et du système primaire.

## VALIDATION DU MODÈLE ENERGYPLUS

La validation, présentée ci-dessous, consiste en la comparaison des résultats obtenus par le logiciel EnergyPlus avec une base de données préenregistrées. La comparaison a été réalisée à deux niveaux (locaux desservis et système VAV). Les paramètres, comparés au niveau des locaux desservis par le système VAV, sont la température ambiante, le débit d'air soufflé et la température de soufflage. Au niveau du système, seul le débit du ventilateur a été utilisé pour les fins de la

comparaison. Le but de cette validation est d'obtenir la variation semblable des paramètres mentionnés précédemment obtenus par la simulation et l'enregistrement. La consommation d'énergie mensuelle ou annuelle, habituellement utilisée dans les validations, n'a pas été prise en considération, car les données enregistrées n'ont pas permis d'isoler ces valeurs pour le système VAV étudié. Même si la base de données d'enregistrement contient beaucoup plus de paramètres, pour les fins de la validation du modèle, seuls les paramètres suivants ont été pris en considération :

- Au niveau des locaux présélectionnés :  
La température ambiante, le débit d'air soufflé, la détection de l'occupation et la température extérieure.
- Au niveau du système :  
La température d'alimentation, le débit du ventilateur, la demande de délestage et la température extérieure.

Le pas d'enregistrement de la base de données est de 1 minute. Pour effectuer la comparaison, le pas de temps de calcul d'EnergyPlus a été fixé à son minimum, soit 10 minutes. La température d'alimentation du système UTA-6 est contrôlée en fonction de la température extérieure et du débit du ventilateur. De plus, les gestionnaires du bâtiment, pour les fins de délestage, modifient les cédules de démarrage des systèmes primaires et secondaires du bâtiment ainsi que les différents points de consignes et séquences de contrôle (ex.: réduction de la vitesse de rotation des ventilateurs, arrêt du refroidisseur d'eau glacée, etc.).

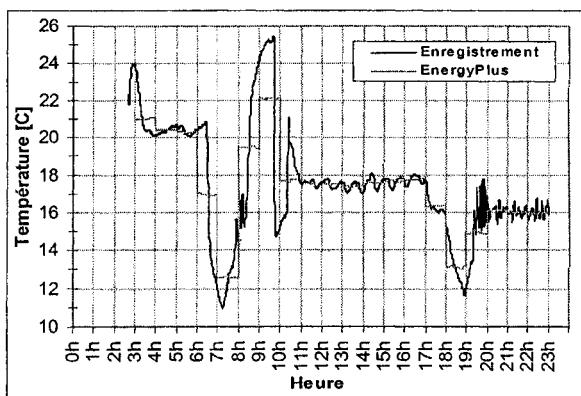


Figure 2. Variation de la température d'alimentation du système UTA-6 le 18 juin 2001.

Par exemple, on remarque sur la figure 2 que la température d'alimentation du système UTA-6 fluctue beaucoup. Celle-ci atteint 11.0°C à 7h19 et 25.4°C à 9h40. Étant donné que la variation de température est

influencée par divers facteurs, une cédule la gérant a été introduite dans le logiciel EnergyPlus. Malheureusement, le pas de temps des cédules pouvant être définie dans EnergyPlus est de 1 heure. La cédule a donc été générée en calculant la température moyenne pour chaque heure d'enregistrement (voir figure 2).

Les figures 3 et 4 présentent respectivement la variation de la température ambiante et du débit volumétrique d'air soufflé dans le local 2304 obtenus par EnergyPlus et par le système d'acquisition de données. On peut constater une bonne concordance des courbes présentant les résultats de simulation et d'enregistrement. Il est à noter que les ajustements furent mineurs pour arriver à ces résultats grâce à un degré de véracité élevé des données relatives aux locaux étudiés.

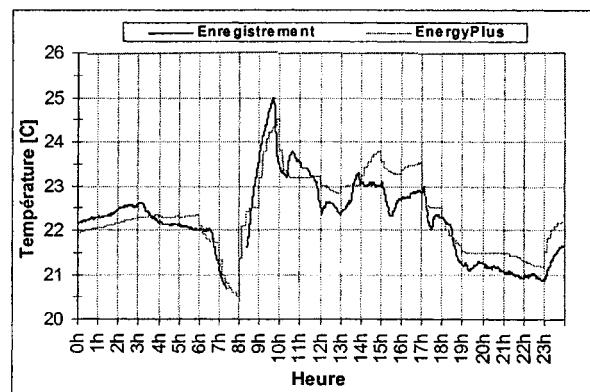


Figure 3. Variation de la température de la salle de cours 2304 le 18 juin 2001.

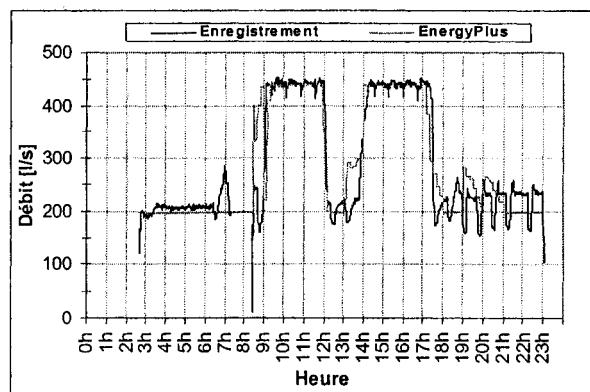


Figure 4. Variation du débit volumétrique d'air soufflé de la salle de cours 2304 le 18 juin 2001.

Cependant, en observant les figures 3 et 4, on note l'impact de la cédule de la température de soufflage, admise comme la moyenne horaire (mentionnée ci-dessus), sur l'allure des courbes obtenues par la

simulation. Elle a pour effet d'augmenter ou de diminuer de façon tardive ou hâtive, par rapport aux données enregistrées, le débit de soufflage de la pièce ainsi que la température ambiante du local.

L'analyse des résultats de simulation de l'opération du système de ventilation ne fut pas si concluante que celle au niveau des locaux desservis. Le faible degré de véracité des informations recueillies pour les locaux aux vocations autres qu'académique ne permet pas de simuler l'ensemble du système avec précision. La validation de 4 zones du bâtiment avec succès nous permet de croire qu'avec un relevé détaillé accompagné d'un indice de véracité de l'ensemble des zones desservies par le système, il serait possible d'obtenir les résultats escomptés.

### COMPARAISON ENTRE ENERGYPLUS ET DOE-2E

Afin de respecter les limites du logiciel DOE-2E et d'obtenir les mêmes conditions d'opération, le modèle de simulation fait sur EnergyPlus et validé dans l'étape précédente a été légèrement modifié et ajusté. À titre d'exemples :

- La température d'alimentation a été fixée à 16°C. Cette température représente la température d'alimentation moyenne enregistrée sur le système UTA-6 pour le mois de juin 2001.
- Les fractions radiatives de la chaleur sensible dégagée par les occupants et les équipements ont dû être adaptées aux conditions prédéfinies par le logiciel DOE-2E.
- Le choix de l'algorithme de calcul du coefficient de convection interne a également été modifié. Trois algorithmes de calculs du coefficient de convection interne sont disponibles dans EnergyPlus. On y retrouve un algorithme simplifié (coefficient fixe), un détaillé (coefficient variable en fonction de l'inclinaison de la paroi et de la différence de température entre la paroi et le fluide en contact avec celle-ci) et un dernier utilisant un modèle de convection mixte et forcé pour la diffusion de l'air par le plafond.

La figure 5 présente la variation de la température ambiante en fonction du type d'algorithme utilisé. On remarque que le choix de l'algorithme a une influence non-négligeable sur les conditions internes du local 2304, et ce, surtout lorsque le local est inoccupé. Il est à souligner que l'algorithme simplifié (le seul disponible dans DOE-2E) a été choisi pour fin de comparaison, tandis que lors de la validation du modèle, l'algorithme

utilisant un modèle de convection mixte et forcé s'est avéré le plus performant des trois algorithmes mentionnés ci-dessus.

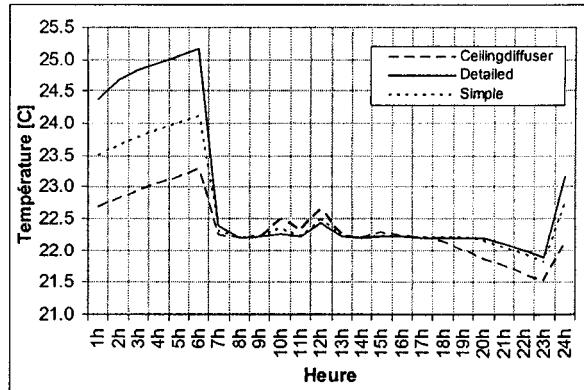


Figure 5. Influence du choix de l'algorithme de calcul du coefficient de convection interne sur la température ambiante du local 2304.

Les figures 6 et 7 présentent les variations des températures et des débits volumétriques d'air soufflé dans le local 2304 obtenus à l'aide d'EnergyPlus et du logiciel DOE-2E. L'analyse des résultats démontre que la différence de température absolue du local entre les deux simulations est en moyenne pour une semaine de 0.24°C. De plus, on remarque que les deux profils journaliers du débit volumétrique sont semblables. La différence absolue moyenne observée pour une semaine entre les débits obtenus à l'aide d'EnergyPlus et de DOE-2E est de 9%.

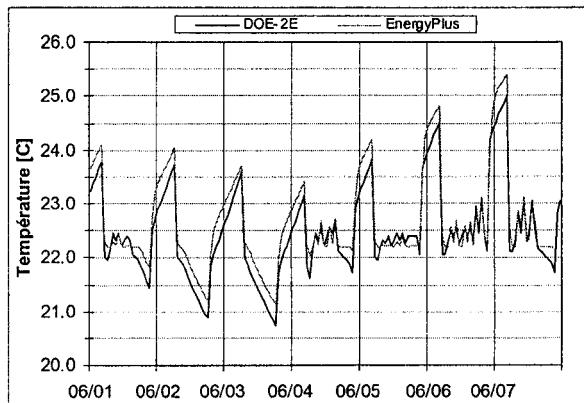


Figure 6. Répartition journalière de la température ambiante de la salle de cours 2304.

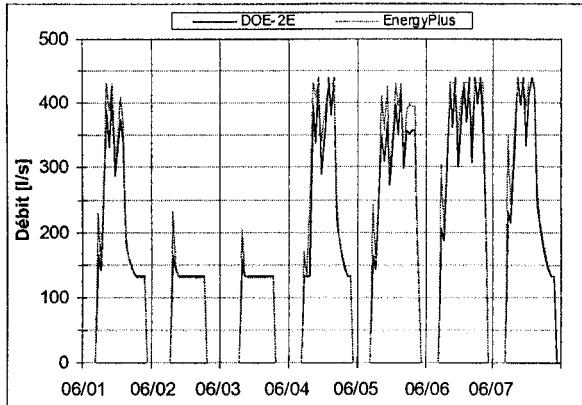


Figure 7. Répartition journalière du débit volumétrique d'air soufflé de la salle de cours 2304.

La principale différence observée porte sur le débit volumétrique d'air soufflé dans la zone lors du démarrage du système. En éliminant la première heure de fonctionnement du calcul de la moyenne, on remarque que l'erreur diminue de 2%. Cette différence lors du démarrage est encore plus visible sur les figures 8 et 9 où l'on présente respectivement la variation du débit volumétrique total du ventilateur et la variation de la demande de refroidissement du système UTA-6, et ce, pour les journées du lundi et du mardi.

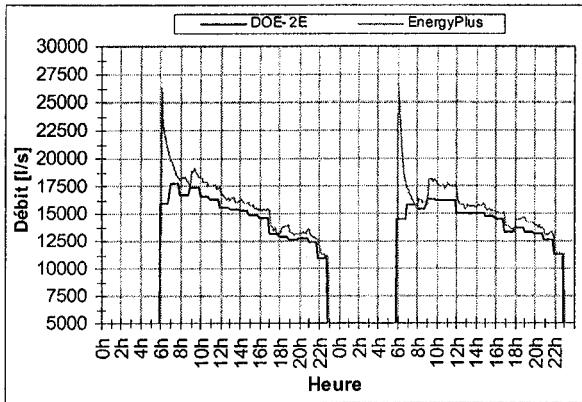


Figure 8. Répartition du débit volumétrique d'air du ventilateur du système UTA-6 pour les journées du lundi et mardi.

Les différences absolues entre les débits et les demandes de refroidissement des simulations sont respectivement de l'ordre de 8.5% et 8.3%. Comme précédemment, si on soustrait la première heure de fonctionnement des systèmes, l'écart entre les débits est plutôt de 5.7% et de 7.9% entre les demandes. Cette différence est partiellement due au pas de temps de 10 minutes utilisé avec EnergyPlus alors que le pas de temps de DOE-2E est de 1 heure.

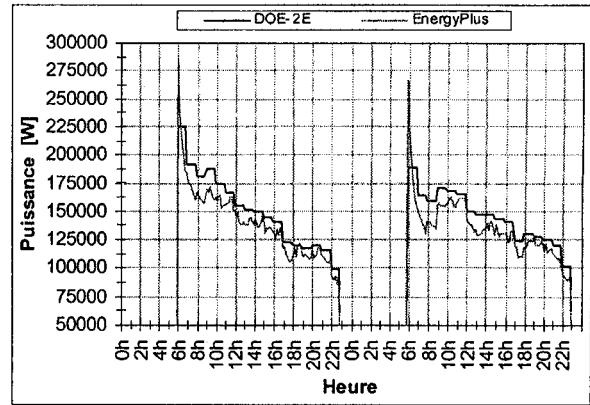


Figure 9. Répartition journalière de la demande de refroidissement au serpentin du système UTA-6 pour les journées du lundi et mardi.

Cependant, il est à noter que selon les données enregistrées sur le système UTA-6, le profil de la variation du débit du système ressemble davantage au profil obtenu à l'aide du logiciel EnergyPlus.

## REMARQUES ET OBSERVATIONS

La présente section est un résumé des principales remarques et observations faites lors de la programmation du modèle dans EnergyPlus et de l'analyse des résultats obtenus. Le but n'est donc pas de faire la liste complète des particularités du logiciel et de ces anomalies, mais de présenter celles observées par notre équipe.

### *Remarques portant sur la modélisation et la simulation*

- 1- L'une des premières observations faites porte sur la quantité de données et de lignes codes à introduire dans EnergyPlus versus DOE-2E. Par exemple, pour simuler le système UTA-6 et les zones qu'il alimente, environ 7500 lignes codes ont été introduites dans DOE-2 contre 22000 lignes codes dans EnergyPlus. Il est à noter, que lors de la modélisation des zones desservies par le système UTA-6, l'ensemble des partitions internes ont été simulées dans les deux logiciels. Dans le logiciel EnergyPlus, une partition interne doit être définie dans les 2 zones contrairement à DOE-2E où l'on doit définir la partition seulement dans une des deux zones. Cette différence augmente ici considérablement le nombre de lignes codes dans EnergyPlus.
- 2- La définition, dans EnergyPlus, d'une paroi (mur, toiture, plancher, fenêtre ou partition) impose l'introduction des coordonnées cartésiennes (x,y,z) de chaque extrémité de celle-ci comparativement à DOE-2E où il est possible d'introduire seulement la superficie, l'inclinaison et l'orientation de la

- paroi. Par contre, l'introduction des coordonnées cartésiennes, grâce à un fichier .dxf créé par EnergyPlus, permet de visualiser les parois décrites dans celui-ci.
- 3- Les outils de conversion du code DOE-2 et des fichiers météorologiques ont été testés avec succès. Le premier outil permet de convertir la section LOAD du code DOE-2 incluant l'ensemble des horaires qui doivent être décrits de façon non-abrégée. Par la suite, l'utilisateur doit inclure les informations manquantes et requises par le logiciel EnergyPlus qui ne les sont pas avec DOE-2. Le second outil permet quant à lui de convertir les fichiers météos standard (ex. WYEC2) en fichier de type .ewp, utilisé par EnergyPlus.
  - 4- Les différentes possibilités d'agencement des éléments d'un système rendent l'utilisation de ce logiciel intéressante lors de la simulation de systèmes dits non-conventionnels.
  - 5- La possibilité d'effectuer le transfert de l'air d'une zone à une autre permet de simuler les conditions ambiantes d'une zone climatisée de façon indirecte. Cette option a été testée avec succès sur les salles de toilettes qui étaient climatisées de façon indirecte par les corridors du 2<sup>ème</sup> étage.
  - 6- La possibilité de contrôler deux systèmes de traitement de l'air permet, par exemple, de modéliser une unité de climatisation de fenêtre avec un système central de traitement de l'air. La séquence de contrôle de ces unités est alors déterminée à partir des priorités en mode chauffage et refroidissement fixées par l'utilisateur.
  - 7- Les simulations ont été lancées sur un ordinateur munis d'un processeur AMD Athlon XP de 1000 Mhz et de 768 Meg. de mémoire vive. Pour la simulation d'un mois, avec un pas de temps de calcul de 10 minutes, le temps de calcul sur cet ordinateur était de 14 minutes sous EnergyPlus comparativement à quelques secondes avec DOE-2 (Pas de temps = 1h).
  - 8- La réduction du pas de temps de calculs permet d'évaluer de façon plus précise l'impact des variations importantes et subites de certains paramètres d'opération et/ou de fonctionnement des systèmes et des zones qu'il alimente. Cependant, les cédules sont de type horaire ce qui réduit la précision des résultats obtenus. En observant les figures 10 et 11, on remarque l'influence du choix du pas de temps sur les demandes de refroidissement et de débit d'air du système de traitement de l'air UTA-6. Ainsi, on peut constater que la réduction du pas de temps, permet de mieux évaluer la dynamique du bâtiment étudié.

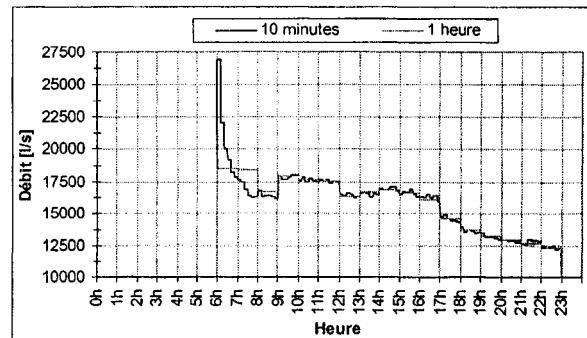


Figure 10. Variation du débit volumétrique d'air du système UTA-6 en fonction du pas de temps.

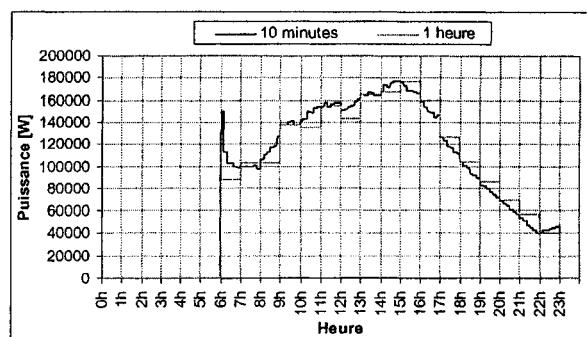


Figure 11. Variation de la demande de refroidissement du système UTA-6 en fonction du pas de temps.

- 9- Le point de lecture du contrôleur de l'humidificateur ne peut être fait que par la lecture de l'humidité d'une zone et non d'un nœud du système de traitement de l'air (ex. : le retour d'air).
- 10- L'utilisation d'un même nœud de lecture pour contrôler deux serpentins, soit un serpentin de chauffage et un serpentin de refroidissement, occasionne des erreurs dans la simulation (un des deux serpentins ne fonctionnera pas).
- 11- L'ordre d'apparition des serpentins dicte l'ordre et l'emplacement des sondes de température (points de lecture). Par exemple, le point de lecture du contrôle de la température d'alimentation en mode refroidissement doit absolument être situé directement à la sortie du serpentin de refroidissement.

#### Anomalies rencontrées

- 1- Une perte de débit massique a été observée entre certains nœuds des systèmes CVCA. La provenance de ces pertes de débit n'a pu être décelée. Pour une journée type, la perte maximale observée a été de 0.20181kg/s, soit 0.83% du débit total. Il est à noter que la quantité d'air perdue varie en fonction du temps.

- 2- Lors de la simulation, le débit maximum du ventilateur d'alimentation du système VAV n'était pas respecté. Ainsi, lorsque la demande au ventilateur central augmentait et dépassait la capacité du ventilateur, le logiciel augmentait malgré tout le débit d'air pour répondre aux besoins de chacune des zones. Le débit volumétrique nominal du ventilateur d'alimentation était de 23000l/s et la somme des débits des boîtes à volume d'air variable était de 27060l/s. Ainsi, le facteur de diversité du système (85%) n'était pas respecté.

## CONCLUSION

L'analyse comparative du logiciel EnergyPlus a permis de démontrer la fiabilité des résultats obtenus et la performance des algorithmes de calculs utilisés par celui-ci. La force du logiciel EnergyPlus est, entre autres, de permettre à l'utilisateur de simuler toutes sortes d'arrangements de systèmes et d'avoir le contrôle sur presque tous les paramètres de la simulation, tels les fractions radiatives et convectives des charges internes, le pas de temps de calculs, etc. Comme il a été démontré pour le démarrage du système, le pas de temps de calculs permet de mieux simuler certaines opérations des systèmes CVCA.

Lors de l'analyse des résultats obtenus avec DOE-2E et EnergyPlus, certaines différences ont été observées entre ceux-ci. Ces différences étaient principalement dues au manque de flexibilité du logiciel DOE-2. Par exemple, l'algorithme de calcul du coefficient de convection interne utilisé par défaut dans le logiciel DOE-2 peut être choisi selon la complexité de calculs dans le logiciel EnergyPlus. Il a été démontré que le choix de cet algorithme avait une influence non-négligeable sur la variation de la température ambiante des zones, et ce, surtout lorsque le système CVCA était en arrêt.

## BIBLIOGRAPHIE

Witte M.J., Henninger R.H., Glazer J. (2001), *Testing and validation of a new building energy simulation program*, Seventh International IBPSA Conference, Rio de Janeiro, Bresil.

Crawley D.B., Winkelmann F.C., Lawrie L.K. Pedersen C.O. (2001), *EnergyPlus: New capabilities in a whole-building energy simulation program*, Seventh International IBPSA Conference, Rio de Janeiro, Bresil.

Henninger R.H., Witte M.J. (2001), *EnergyPlus Testing with HVAC BESTEST*, The Ernest Orlando Lawrence Berkeley National Laboratory.

Henninger R.H. and Witte M.J. (2001), *EnergyPlus Testing with ANSI/ASHRAE Standard 140-2001 (BESTEST)*, The Ernest Orlando Lawrence Berkeley National Laboratory.

*Input Output Reference – The Encyclopedic Reference to EnergyPlus Input and Output*, The Ernest Orlando Lawrence Berkeley National Laboratory, Juin 2001.

*EnergyPlus Engineering Document – The reference to EnergyPlus Calculations*, The Ernest Orlando Lawrence Berkeley National Laboratory, Juin 2001.

*DOE-2 Engineers Manual*, Lawrence Berkeley Laboratory and Los Alamos National Laboratory, November 1982.

## BIBLIOGRAPHIE

ASHRAE 2001. *Handbook of Fundamentals*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. Atlanta, GA, USA.

*Code modèle national de l'énergie pour les bâtiments – Canada 1997*. Commission canadienne des codes du bâtiment et de prévention des incendies, Conseil national de recherches du Canada.

Crawley, D.B., Winkelmann, F.C., Lawrie, L.K., Pedersen C.O. (2001). *EnergyPlus: New capabilities in a whole-building energy simulation program*. Seventh International IBPSA Conference, Rio de Janeiro, Bresil.

*DOE-2 Engineers Manual*. Lawrence Berkeley Laboratory and Los Alamos National Laboratory, November 1982.

*DOE-2 Reference Manual, Version 2.1E*. Lawrence Berkeley Laboratory and Los Alamos National Laboratory, November 1993.

*DOE-2 Supplement, Version 2.1E*. Lawrence Berkeley Laboratory and Los Alamos National Laboratory, November 1993.

*EnergyPlus Engineering Document – The reference to EnergyPlus Calculations*. The Ernest Orlando Lawrence Berkeley National Laboratory, Juillet 2002.

*EnergyPlus Programming Standard*. The Ernest Orlando Lawrence Berkeley National Laboratory, Avril 2002.

Henniger, R.H., Witte, M.J. (2001). *EnergyPlus Testing with HVAC BESTEST*. The Ernest Orlando Lawrence Berkeley National Laboratory.

Henniger, R.H., Witte, M.J. (2001), *EnergyPlus Testing with ANSI/ASHRAE Standard 140-2001 (BESTEST)*. The Ernest Orlando Lawrence Berkeley National Laboratory.

Incorpera, F.P., De Witt, D.P., *Fundamentals of Heat and Mass Transfer* (5e éd.). John Wiley & Son, 2001.

*Input Output Reference – The Encyclopedic Reference to EnergyPlus Input and Output*. The Ernest Orlando Lawrence Berkeley National Laboratory, Juillet 2002.

McQuiston, F.C., Parker, J.D. *Heating, ventilating and air conditioning. Analysis and design* (3e éd.). New-York : John Willey & Sons.

Strand, R.K., Buhl, W.F., Lawrie, L.K. *Guide for Module Developers – Everything You Need to Know about EnergyPlus Calculational Development*. The Ernest Orlando Lawrence Berkeley National Laboratory, Avril 2002.

Witte, M.J., Henninger, R.H., Glazer J. (2001). *Testing and validation of a new building energy simulation program*. Seventh International IBPSA Conference, Rio de Janeiro, Bresil.