



HAL
open science

Intégration à TRNSYS du noyau de CODYRUN, code de simulation thermo-aéraulique de bâtiments: le Type 59

Alain Bastide, Harry Boyer, Philippe Lauret, Franck Lucas, François Garde

► To cite this version:

Alain Bastide, Harry Boyer, Philippe Lauret, Franck Lucas, François Garde. Intégration à TRNSYS du noyau de CODYRUN, code de simulation thermo-aéraulique de bâtiments: le Type 59. 4ième séminaire TRNSYS-EES, Sep 2001, Lyon, France. pp.1-6. hal-00768184

HAL Id: hal-00768184

<https://hal.science/hal-00768184>

Submitted on 21 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Intégration à *TRNSYS* du noyau de *CODYRUN*, code de simulation thermo-aéraulique de bâtiments : le Type 59

Alain BASTIDE, Harry BOYER, Philippe LAURET, Franck LUCAS, François GARDE

Université de La Réunion
Laboratoire de Génie Industriel
Equipe Génie Civil-Thermique de l'Habitat
40, Avenue de Soweto, 97410 SAINT PIERRE La RÉUNION
alain.bastide@univ-reunion.fr, harry.boyer@univ-reunion.fr

Résumé

Ce papier traite de l'intégration à l'environnement TRNSYS du noyau de calcul d'un modèle développé depuis une dizaine d'années à l'Université de la Réunion, développement entrepris initialement en partenariat avec le CETHIL de l'INSA de Lyon. Il part du constat d'une segmentation des utilisateurs-développeurs en thermique des bâtiments, partagés entre un groupe sous environnement TRNSYS (type19 et 56 principalement) et l'autre mettant en œuvre des codes spécifiques (ESP, APACHE, BUS, CODYRUN, ...). Chacun de ces choix possède des avantages et/ou inconvénients, le bénéfice de l'environnement TRNSYS étant principalement celui de la ré-utilisabilité des modules construits par la communauté.

1. Historique et fonctionnement global

CODYRUN est un code de simulation de bâtiments multizones, intégrant des aspects aérauliques et hydriques. Globalement, il comprend trois modules principaux, thermique, aéraulique et hydrique, couplés itérativement à chaque pas de temps. Leurs principales caractéristiques sont les suivantes :

- Module thermique
 - Couplage itératif des zones
 - Transferts par conduction à l'aide de l'analogie thermo-électrique
 - Echanges superficiels linéarisés
 - Résolution par différences finies
- Module aéraulique
 - Modèle en pression intégrant les petites ouvertures (cracks) et la VMC idéalisée.
 - Prise en compte des grandes ouvertures intérieures verticales à l'aide du modèle de WALTON.
 - Prise en compte des grandes ouvertures extérieures par utilisation du modèle de WALTON et superposition des pressions liées au vent et au tirage thermique.
 - Résolution du système non linéaire obtenu par Newton-Raphson.
- Module hydrique
 - L'humidité est transportée par les débits.
 - Intégration d'un tampon hygroscopique à teneur en eau du cœur variable.
 - Résolution par différence finies des équations de bilan de poids d'eau par zone.

L'idée initiale de son développement (1990) était celle d'un outil à orientation aide à la conception et recherche, recourant alors à une notion de logiciel multi-modèle. Les aspects de détail liés à ce code de simulation sont explicités dans les références [1-3]. D'autres aspects ont trait à sa validation expérimentale [4-7] et au couplage avec des systèmes de traitement d'air [8] ou aux applications en conception [9-10].

Techniquement parlant, il s'agit d'un logiciel développé initialement en C sous l'environnement Windows, à l'aide de Quick C/ Windows, compilateur 16 bits. Des problèmes de compatibilité étant intervenues avec les versions successives de Windows (par exemple gestion défaillante par Windows'98 de la mémoire allouée par des applications 16 bits), nous avons alors séparé le programme en deux applications distinctes, communiquant par fichiers et/ou pointeurs. Ces parties traitent ainsi de la description du bâtiment (à l'aide de fenêtres des saisies) pour CODYDESK et du noyau de calcul thermo-hygro-aéraulique, à savoir CODYKERN.

Notre travail a donc été de séparer dans notre code de calcul la partie descriptive de la partie calculs.

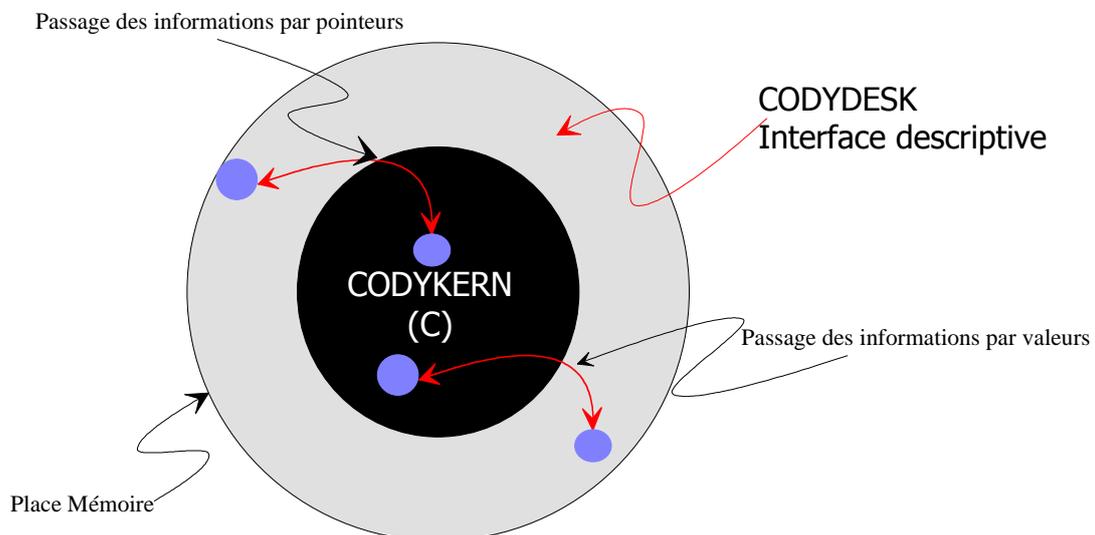


Figure 1 : Fonctionnement de CODYRUN

Les informations entre les deux parties peuvent transiter grâce à deux méthodes :

- La méthode par pointeur. Les pointeurs sont définis et initialisés au démarrage de CODYRUN, puis CODYKERN et CODYDESK partagent des emplacements mémoires identiques. L'intérêt d'une telle méthode est d'accroître les performances du code et de minimiser l'allocation de la mémoire.
- La méthode par valeur. Lorsque des données sont faiblement utilisées, il est parfois plus commode d'utiliser cette méthode moins difficile à mettre en œuvre que la précédente. Mais lorsque que le flux d'informations est trop important, le temps global de calcul est alors fortement augmenté.

Donc dans le Type 59, dans un souci de performances, toutes les informations transitent par pointeur.

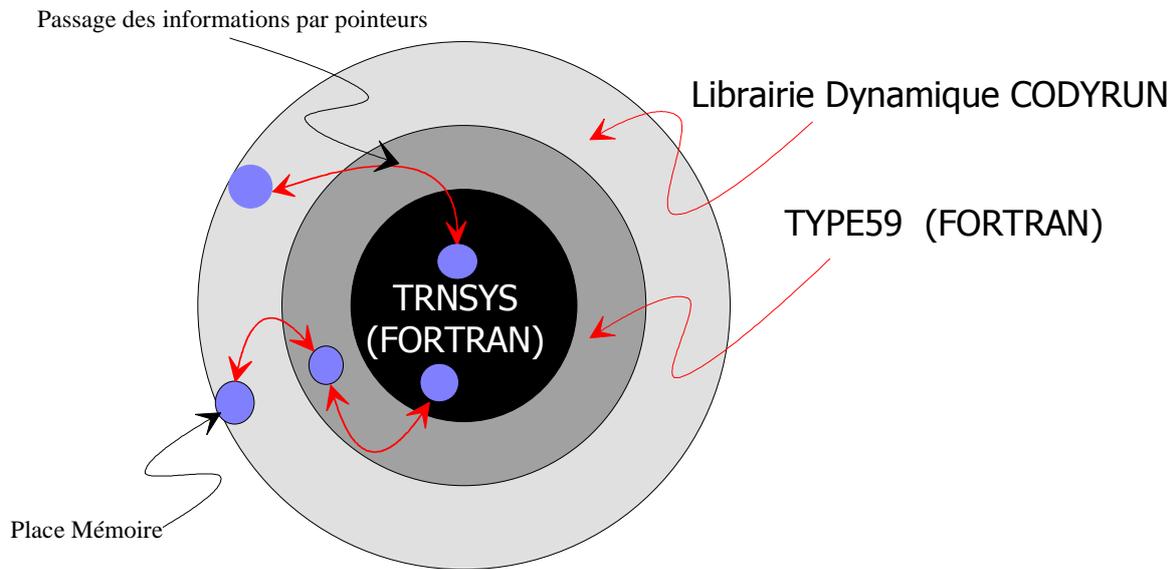


Figure 2 : Fonctionnement du Type 59

Au démarrage de ce travail de portage de l'application sous TRNSYS, le noyau de calcul a été recompilé sous la forme d'une *Dynamic-Link Library* (CODYDLL). Cette librairie dynamique donne quatre accès à ses ressources par l'intermédiaire de fonctions. Trois de ces fonctions permettent de préparer la simulation en initialisant tous les pointeurs et variables, puis une fonction est appelée à chaque pas de temps pour la résolution du système, enfin la troisième fonction libère la mémoire. La dernière fonction permet la récupération des erreurs internes à notre DLL pour qu'elles apparaissent dans le fichier résultat de simulation de TRNSYS (List File). Des informations de débogage de la librairie peuvent aussi circuler pour vérifier son bon fonctionnement.

D'un point de vue plus pratique, la recompilation de la DLL a posé le problème de compatibilité inter-compilateur. En effet, CODYRUN est écrit en langage C et est compilé avec Microsoft Quick C/Windows, compilateur qui n'a pas été maintenu par la suite (pas de compatibilité de l'API Windows avec la série des compilateurs suivants, à savoir Visual C++ 1.0,...). Le CODYKERN est lui, compilé sous Borland C++ (pour des raisons de simplicité d'utilisation des outils de débogage, *CodeGuard* en particulier). Enfin, la DLL doit être compilée pour des raisons de compatibilité avec TRNSYS avec le compilateur Microsoft C++. Notons qu'au passage des incorrections de programmation ont été décelées dans les bibliothèques de fonctions mathématiques utilisées par le noyau de calcul, bibliothèques pourtant largement diffusées (Numerical Recipes).

Par la suite, un autre problème du aux exigences du compilateur 32-bit est apparu et une partie de la librairie a été réécrite (problème de format de fichier lié à des types de bases ne possédant par la même taille mémoire entre les différents environnements).

De plus, il importe de comprendre que CODYRUN dans sa version initiale ne fonctionne pas comme un type TRNSYS, car il possède son propre lecteur de données météorologiques, son lecteur de fichier bâtiment, ses modèles de traitement d'air, sa gestion de charges internes, tous ces aspects étant encapsulés dans le code. De même, un solveur (résolution de systèmes linéaires et non linéaires) est aussi intégré à l'application.

Nous avons donc créé un Type spécifique pour lire les fichiers de données météorologiques de CODYRUN. Ce type (Type 159) permet la lecture des données météorologiques et l'adaptation des sorties en fonction du pas de temps de la simulation. Pour les systèmes de traitement d'air et pour les charges internes, nous avons pris le choix de définir des entrées au Type telles que la puissance sensible, la puissance latente, et la position zonale des charges. Le nom du fichier bâtiment est lui, un paramètre du Type 59, il est passé à la DLL par l'intermédiaire du code fortran. Les paramètres servent à définir le type d'initialisation souhaité pour le bâtiment, ce en température et en humidité ou encore le type de modèle en humidité (avec ou sans tampon,...) et les paramètres associés.

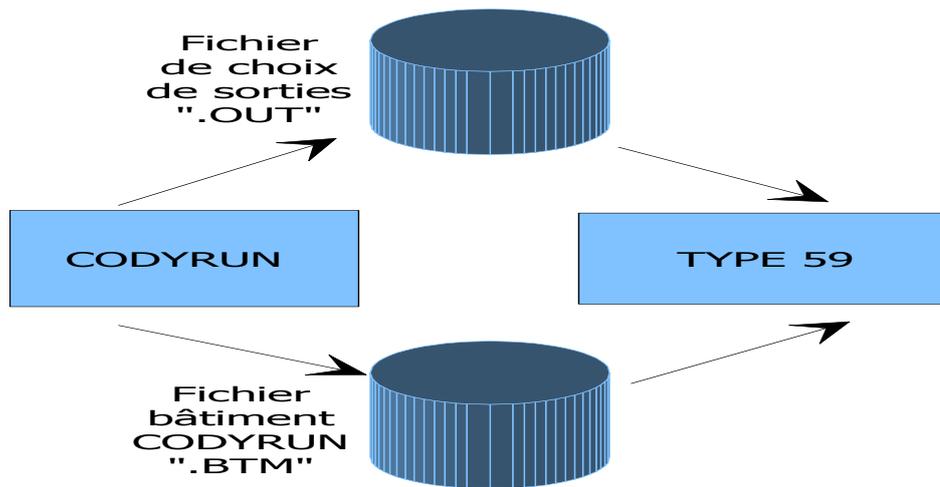


Figure 3 : Système de fichier du Type 59

Des améliorations sont en cours d'élaboration sur l'aspect fonctionnel du Type 59 et plus particulièrement sur la possibilité d'offrir à l'utilisateur le choix des sorties qu'il souhaite analyser. Cette dernière amélioration se fait par l'intermédiaire d'un fichier dit « neutre » (mode texte) créé soit depuis CODYDESK (automatiquement et par défaut) soit directement par l'utilisateur suivant une syntaxe simple et compréhensible. Les types de sorties disponibles à l'heure actuelle sont de l'ordre d'une trentaine (températures caractéristiques de zones, de parois, débits d'air, puissances,...)

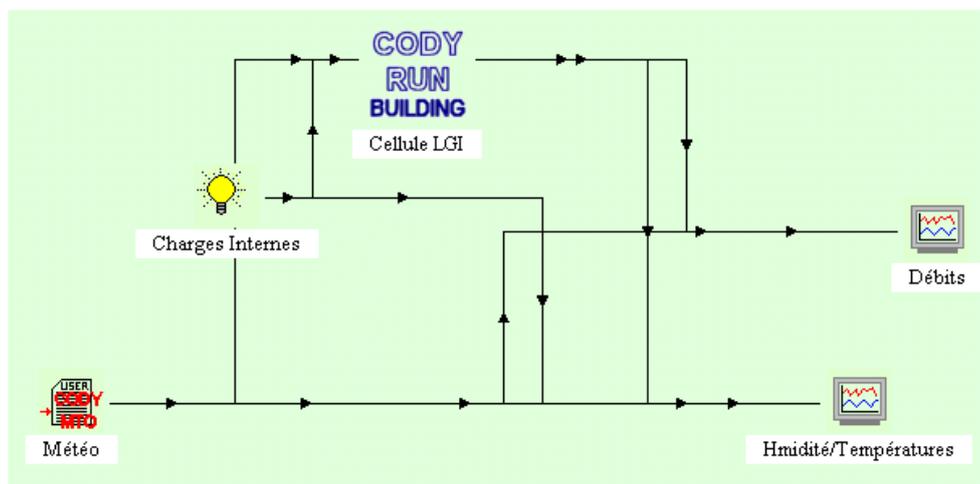


Figure 4 : Implémentation du Type 59 dans l'environnement TRNSYS

2. Quelques caractéristiques intéressantes du type 59

- Le code résout à la fois la thermique, l'aérodynamique et l'hydraulique, avec cependant les restrictions et simplifications propres à chacune des parties (hypothèses, phénomènes intégrés, composants disponibles, mode de couplage choisi,...). Il n'y a donc qu'une seule interface (CODYDESK) pour créer un bâtiment englobant tous ces aspects.
- Le Type 59 fonctionne grâce à l'analogie thermique/électricité. Les parois sont discrétisées par des méthodes analogiques, tels que le modèle couche par couche, le 3R2C ou le R2C
- Le couplage enveloppe/système a été validé pour des pas de temps de la minute. Il est donc possible de coupler des systèmes de traitement d'air fonctionnant avec ce pas de temps.
- La modélisation des phénomènes thermiques a été validée expérimentalement.

3. Perspectives

- Lorsque l'interface a été écrite la première fois en Quick C, les outils d'aide à l'agencement des fenêtres, à l'écriture du code, n'étaient alors que très peu développés. C'est pourquoi, l'interface n'est pas des plus ergonomiques. Néanmoins, une interface 32-bit jouissant de toutes les fonctions utiles à la conception de bâtiment verra le jour d'ici quelques mois.
- Le fichier de description du bâtiment est propriétaire. Nous sommes donc en train de construire une passerelle entre les fichiers CODYRUN et SimCad, ceci pour permettre facilement le portage des bâtiments entre les Type 59 et 56.
- Le temps de calcul du Type 59 observé lors des simulations comparatives (en mode thermique) est supérieur à celui du Type 56. Cependant, aucun travail spécifique n'a été mené dans ce sens.
- Dans CODYRUN, les conditions initiales thermiques du bâtiment sont obtenues en faisant une pré-simulation sur quelques jours (initialisation). Dans le Type59, les données météorologiques étant extérieures, cette démarche n'a pu être reproduite et les résultats des premiers jours de simulation sont influencés par les conditions initiales du schéma de résolution. L'utilisateur doit donc en avoir conscience lors de l'analyse des résultats.

Conclusion

En dehors des aspects techniques de programmation parfois assez ardues, le portage de CODYRUN vers un type TRNSYS constitue pour l'équipe un tournant qui finalise une dizaine d'années de recherche. Ainsi, les travaux menés vis à vis de l'amélioration du modèles, sa validation, l'ajout de nouveaux composants, ... sont en quelque sorte capitalisés au sein du type 59. Il fonctionne actuellement dans une version prototype en interne à l'équipe. Ce type assez complémentaire par rapport à l'existant devrait tirer avantage des développements entrepris autour du type 56 et permettra une fois ce développement finalisé d'apporter à la communauté des thermiciens du bâtiment une nouvelle possibilité de modélisation de bâtiments multizones sous TRNSYS.

Remerciements

Les auteurs tiennent à remercier le Professeur Alain CORDIER du LESETH pour ses conseils avisés et ses encouragements dans ce travail. De même, le Professeur Jean BRAU est depuis l'origine de CODYRUN celui qui n'a eu de cesse de nous convaincre d'emprunter la voie TRNSYS. Ces efforts n'ont pas été vains.

Références

- [1] H. Boyer, J. P. Chabriat, B. Grondin-Perez, C. Tourrand & J. Brau : *Thermal Building Simulation and Computer Generation of Nodal Models*, **Building and Environment**, vol. 31, N°3. pp. 207-214, 1996.
- [2] H. Boyer, A.P. Lauret, L. Adelard, T.A. Mara
Building ventilation : a pressure airflow model computer generation and elements of validation
Energy and Buildings, 29 (1999), 283-292
- [3] H. Boyer, F. Garde, J.C. Gatina, J. Brau
A multi model approach of thermal building simulation for design and research purposes
Energy and Buildings, 28 (1998), 1, pp. 71-79
- [4] F. Lucas, T. Mara, F. Garde, H. Boyer
A comparison between CODYRUN and TRNSYS, simulation codes for thermal building behaviour
Renewable Energy, 15 (1998), Part. II, 1624-1633
- [5] T. Soubdhan, T.A. Mara, H. Boyer, A. Younès : *Use of BESTEST procedure to improve a building thermal simulation program*, **Renewable Energy**, pp. 1800-1803 Part III, 1-7 July 2000.
- [6] T. A. Mara, F. Garde H. Boyer, M. Mamode
Empirical Validation of the Thermal Model of a Passive Solar Cell test
Energy and Buildings, Volume 33, 6, 2001, pp. 589-599.
- [7] A.J.P. Lauret, T.A. Mara, H. Boyer, L. Adelard, F. Garde
A validation methodology aid for improving a thermal building model: case of diffuse radiation accounting in a tropical climate
Energy and Buildings, vol. 33, 7, pp 711-718, 2001
- [8] F. Garde, F. Lucas, H. Boyer, J. Brau
Multiple model approach of a residential heat pump for integration in building thermal simulation code and comparison with experimental results, **ASHRAE Transactions**, Volume 107, Part 1, No 4412, Janv. 2000
- [9] F. Garde, H. Boyer, J.C. Gatina
Demand side management in tropical island buildings. Elaboration of global quality standards for natural and low energy cooling in buildings
Building and Environment, 34 (1998), 1, pp. 71-84
- [10] F. Garde, T. Mara, A.P. Lauret, H. Boyer, R. Celaire
Bringing simulation to implementation : presentation of a global approach in the design of passive solar buildings under humid tropical climates, **Solar Energy**, vol. 71, 2, 109-120, 2001
- [11] Solar Energy Laboratory
TRNSYS 15.0 Manual, University of Wisconsin-Madison