

AUTOMATISCHE GENERIERUNG VON MODULAREN DYNAMISCHEN GEBÄUDEMODELLEN

Jürgen Schumacher, Eric Duminił, Ursula Eicker
 Hochschule für Technik Stuttgart
 Schellingstrasse 24, 70174 Stuttgart, Germany
 juergen.schumacher@hft-stuttgart.de

KURZFASSUNG

Modulare Konzepte für thermische Gebäudesimulation sind selten, da durch langwelligen und konvektiven Strahlungsaustausch die Oberflächen- und Lufttemperaturknoten gekoppelt sind. Die gängigen Gebäudesimulationsverfahren lösen simultan ein linearisiertes Gleichungssystem, in denen alle Temperaturknoten energetisch bilanziert werden. In der vorliegenden Arbeit wurde konsequent ein modulares Konzept für die Gebäudesimulation entwickelt, in welchem jedes Bauteil als eigener Simulationsblock beispielsweise zur numerischen Lösung der Wärmeleitungsgleichung konzipiert wurde.

Weiterhin wurde eine Schnittstelle zwischen Google SketchUp und der Simulationsumgebung INSEL entwickelt, welche eine automatische dynamische und modulare Gebäudemodellerstellung aus architektonischen Zeichnungen ermöglicht.

EINFÜHRUNG

Modulare Konzepte für thermische Gebäudesimulation sind selten, da durch langwelligen und konvektiven Strahlungsaustausch die Oberflächen- und Lufttemperaturknoten gekoppelt sind. In der vorliegenden Arbeit wurde konsequent ein modulares Konzept für die Gebäudesimulation entwickelt, in welchem jedes Bauteil als eigener Simulationsblock zur Lösung der Wärmeleitungsgleichung konzipiert wurde. Weitere Blöcke lösen die konvektive Energiebilanz des Raumlufknoten, den langwelligen Strahlungsaustausch, sowie den kurzwelligen Strahlungsdurchgang durch Verglasungen etc. Die Implementierung der Blöcke erfolgte in der Simulationsumgebung INSEL (www.insel.eu). Da durch die Modularität der Simulationsblöcke jedes Bauteil viele Verknüpfungen der Inputs und Outputs aufweist, ist eine händische Erzeugung des gesamten Gebäudemodells in graphischen oder Texteditoren kaum möglich.

In der vorliegenden Arbeit wurde eine Schnittstelle zwischen Google SketchUp und der Simulationsumgebung INSEL entwickelt, welche eine automatische dynamische und modulare Gebäudemodellerstellung ermöglicht.

Der Algorithmus erkennt auch für größere Wohngebiete automatisch die Einzelgebäude und generiert für jedes Gebäude ein dynamisches Einzonenmodell.

Die entwickelten numerischen Modelle wurden sowohl mit der VDI Gebäudesimulationsrichtlinie 6020 als auch mit Messwerten aus Bürobauprojekten validiert.

SIMULATIONSUMGEBUNGEN

SketchUp

SketchUp (<http://SketchUp.google.com/>) ist eine Software zur Erstellung von 3D-Entwürfen in Architektur und Design. Die einfache Version ist kostenlos, und kann z.B. für Forschung benutzt werden.

Um die Gebäudegeometrie mit SketchUp zu zeichnen, kann man entweder Autocad Pläne importieren und extrudieren, CityGML Dateien konvertieren oder von Grund auf ein Entwurfsmodell bauen.

Ein großer Vorteil von SketchUp ist, dass die ganze Software mit Ruby skriptfähig ist.

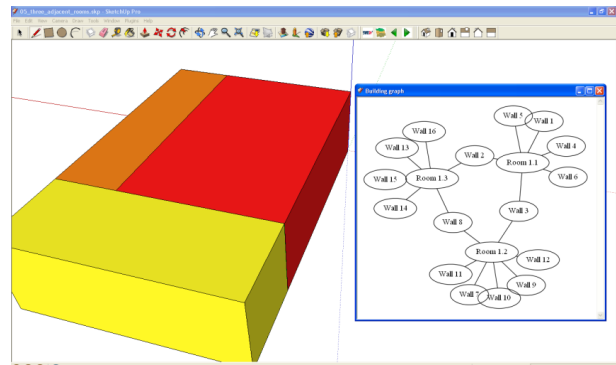


Abbildung 1 Google Sketch-up Modell für drei thermische Zonen mit automatisch generierter Struktur des modularen Simulationsmodells

Ruby

Ruby ist eine Skriptsprache, mit der Simulationsprozesse und Parametervariation sehr flexibel gesteuert werden können.

Insel

Die modulare Simulationsumgebung INSEL (Integrated Simulation Environment Language) wurde in den 1980er Jahren an der Universität Oldenburg entwickelt. Hauptanwendungsbereich der Software waren die Themen Photovoltaik und Energiemetereologie. Seit Anfang der 2000er Jahre wird INSEL an

der Hochschule für Technik Stuttgart weiterentwickelt und um die Bereiche Solarthermie – insbesondere solares Heizen und Kühlen – sowie Gebäudesimulation erweitert.

Technisch ausgedrückt ist INSEL ein kausales Blockdiagramm-basiertes Simulationssystem. Es unterscheidet sich von ähnlichen Programmen – wie zum Beispiel TRNSYS, MATLAB & Simulink oder Modelica – zum einen durch die große Anzahl vorhandener Bausteine zur Simulation erneuerbarer Energiesysteme und zum anderen durch den Lösungsansatz für komplexe Simulationsmodelle. INSEL verwendet hierzu keinen universellen Solver für algebraische und Differenzialgleichungen, sondern Blöcke, die explizite Gleichungszusammenhänge enthalten. Ein Sortieralgorithmus bringt die Modelle in eine einzügige Reihenfolge, die sowohl lokale Schleifen als auch lokale Verzweigungen enthalten kann. Damit ist INSEL eine universell strukturierte Programmiersprache, die eine Schnittstelle zur Erweiterung der Blockbibliothek zur Verfügung stellt.

Die Einschränkung auf die Kausalität der vorhandenen Blöcke wird zur Zeit bearbeitet, so dass in Zukunft auch nicht-kausale Modellteile in INSEL Modellen verwendet werden können. Dies ist insbesondere im Hinblick auf die Simulation von Wärmenetzen, aber auch für die Simulation komplexer elektrischer Schaltungen – wie zum Beispiel Photovoltaik Generatoren, die auf Verschattungsprobleme untersucht werden sollen – interessant.

Gebäudesimulationsblöcke

Heiz- und Kühllasten von Gebäuden werden in Simulationsverfahren meist über dynamische Modelle aus dem thermischen Antwortverhalten der Gebäudemasse berechnet. Dieses komplexe Problem kann in zwei Betrachtungen separiert werden:

- Die Analyse der Wärmeleitung und –speicherung in individuellen Komponenten (wie Wände, Decken und Fußböden usw.)
- Die Analyse der Interaktion zwischen der Gebäudemasse und der Lufttemperatur über konvektive und langwellige Strahlung, Gewinne durch Solarstrahlung, interne Lasten und Belüftung.

Die Wärmeleitungsgleichung in massiven Wänden als Funktion der Randbedingungen auf beiden Seiten der Wand wird häufig mit der Transfer Funktionen Methode gelöst (z. B. TRNSYS, DOE, EnergyPlus etc.). Die Berechnung der Koeffizienten der Transfer Funktion ist mathematisch anspruchsvoll. Sind sie jedoch bekannt, so können Wärmeströme und Temperaturen sehr effizient analytisch aus den Werten vorheriger Zeitschritte berechnet werden. Eine explizite numerische Lösung der Wärmeleitungsgleichung in jedem Zeitschritt ist möglich, erfordert jedoch einen relativ hohen Rechenaufwand pro Zeitschritt. Stabilität der Lösung kann erreicht werden, wenn die

räumliche und zeitliche Auflösung an materialabhängige Parameter gekoppelt wird. Implizite numerische Lösungen sind stabil, führen aber auf große Gleichungssysteme, die für jeden Zeitschritt simultan gelöst werden müssen.

In den gängigen Simulationsprogrammen sind die einzelnen Rechenverfahren in komplexe Module eingebettet (z.B. TRNSYS Type 56), so dass eine Modifikation von Teilmodellen für den Anwender praktisch nicht möglich ist (Ausnahme z.B. Modelica). Wir verfolgen einen neuen, modularen Ansatz, der auf einzelnen Blöcken basiert, die beliebig zu komplexeren Gesamtmodellen kombinierbar sind. Die für eine dynamische Gebäudesimulation wichtigsten sind ein Block, in dem der Strahlungsaustausch zwischen Umgebungsflächen berechnet werden kann, ein Block für die Bilanzierung von Luftknoten sowie ein Block für die Lösung der Wärmeleitungsgleichungen innerhalb von Wandaufbauten. Im Gegensatz zu Ansätzen, wie sie zum Beispiel in Modelica verfolgt werden, werden die Teilmodelle nicht durch einen Satz von Differenzialgleichungen angegeben, die dann ein übergreifender Solver zeitschrittweise lösen muss, sondern alle Komponentenmodelle sind in expliziter Form ausprogrammiert, so dass an den Ausgängen der Einzelkomponenten Temperaturen und Wärmeströme abgegriffen werden können.

Langwelliger Strahlungsaustausch (Block RADI)

Um den Strahlungsaustausch zwischen den Wandoberflächen einer Zone berechnen zu können, muss die Eigenemission jeder Fläche mit der langwelligen Absorption aller übrigen Flächen bilanziert werden.

$$q_{r,j} = \underbrace{\varepsilon_j \sigma T_j^4}_{\text{Eigen-Emission Oberfläche } j} - \varepsilon_j \sum_{i=1}^N \Phi_{i,j} \left(\underbrace{\varepsilon_i \sigma T_i^4}_{\text{Eigen-Emission Oberfläche } i} + \underbrace{(1-\varepsilon_i) \bar{f}}_{\text{Reflektierte mittlere Intensität}} \right)$$

Dabei ist die mittlere langwellige Strahlungsintensität innerhalb der Zone näherungsweise:

$$\bar{f} = \frac{\sum_{j=1}^N \varepsilon_j \sigma T_j^4 A_j}{\sum_{j=1}^N \varepsilon_j A_j}$$

Der Nettostrahlungsfluss jeder Oberfläche wird vom Block RADI berechnet. Als Eingangsgrößen verwendet der Block die typischerweise sechs Oberflächentemperaturen der Raumumschließungsflächen und berechnet die Strahlungsflüsse über die Einstrahlzahlen. Parameter des Blocks sind Länge, Breite und Höhe der Zone sowie die einzelnen Emissivitäten der Oberflächen.

Konvektive Bilanz (Block ROOM)

Der ROOM Block ist für die Bilanzierung der konvektiven Wärmeströme zu bzw. von den Umschließungsflächen der Zone sowie eventueller zusätzlicher konvektiver Heiz- oder Kühlenergien zuständig. Berechnet wird die Lufttemperatur sowie die resultierenden konvektiven Wärmeströme an die Umschließ-

sungsflächen. Parameter des Blocks sind das Luftvolumen sowie die Flächen der einzelnen Wände und ein Startwert für die Temperatur des Luftknotens.

Wärmeleitungsgleichung (Block WALL/WALLX)

Die eindimensionale Wärmeleitungsgleichung jeder einzelnen Wand wird explizit in einem eigenen Block berechnet. Dabei wird zwischen innenliegenden Wänden (Block WALL) und aussenliegenden Wänden (WALLX) unterschieden. Beliebige Schichtaufbauten einzelner Wände können über Parameter Quadrupel der Form (Schichtdicke, Wärmeleitfähigkeit, Dichte und Wärmekapazität) berücksichtigt werden. Weitere Parameter des Blocks sind die Fläche der Wand sowie deren beidseitige Koeffizienten für den konvektiven und radiativen Wärmeübergang. Als Eingangsgrößen verwendet der Block neben der Simulationszeit die Lufttemperaturen der beiden angrenzenden Zonen sowie beidseitig solare Gewinne, konvektive Wärmeströme vom Luftknoten, langwellige Wärmeströme von den Umschliessungsflächen (aus Block RADI) sowie eventuell Strahlungsflüsse aus internen Quellen.

Da die Wärmeströmbilanzen für jede Schicht unter Einhaltung des Stabilitätskriteriums vollständig aus Werten des vorherigen Zeitschrittes berechnet werden, können so die Oberflächentemperaturen der Wände explizit berechnet werden ohne dass dazu noch ein externer Solver benötigt wird.

SIMULATIONSVORGANG

Erkennungsalgorithmus (SketchUp + Ruby)

Typische CAD Programme werden oft für Design und Darstellung benutzt, sie sind aber in der Regel weniger für numerische Simulationen geeignet. Das Ziel von unserem Prozess ist zu ermöglichen, direkt aus dem CAD Programm eine dynamische Simulation durchführen zu können.

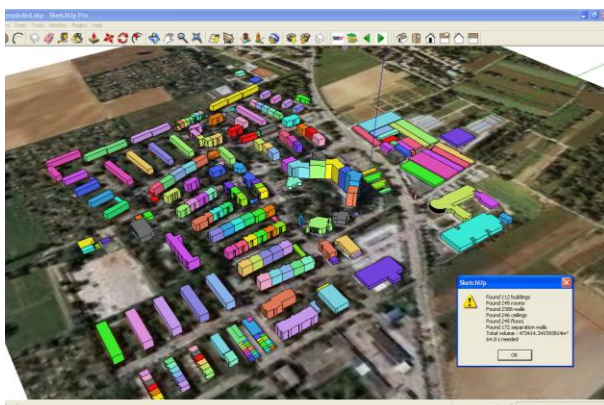


Abbildung 1: Bestands-Wohnbaugebiet mit automatischer Modellerzeugung

Geometrie

Die SketchUp Ruby-API ermöglicht, alle Geometriedaten als Ruby Objekte importieren.

Ein Modell besteht aus mehreren Flächen (Face), die Polygone sind und Wände abbilden. Jede Fläche

entsteht aus mehreren Kanten (Edge), und jede Kante entsteht aus 2 Ecken (Vertex).

Viele Methoden sind schon in SketchUp/Ruby implementiert, so dass der Benutzer leicht abfragen kann, wo die verschiedenen Ecken sind.

Es ist dann leicht, mit Ruby eine Datei zu schreiben (z.B. .xml, .csv oder .insel), die diese Information enthält, um sie zu anderen Programmen zu exportieren.

Topologie

Für die Simulation werden nicht nur geometrische Information benötigt, sondern auch topologische, wie z.B.:

- zu welchem Gebäude gehört diese Wand?
- Ist diese horizontale Fläche ein Kellerdecke, ein Flachdach oder ein Boden?
- Wie viele Zonen teilen dieses Gebäude?
- Welche Wände sind Außenwände?
- Welche Wände sind angrenzend zueinander?
- ...

Um diese Fragen zu beantworten werden zusätzliche Algorithmen und Methoden benötigt, die man mit Ruby schreiben muss:

- Ein Gebäude wird z.B. so erkannt, dass alle Fläche/Kanten/Ecken verbunden sind.
- Eine Fläche die vertikal ist ($\pm 10^\circ$) ist eine Wand.
- Aus verschiedenen horizontalen Flächen, die übereinander liegen, ist die unterste die Kellerdecke, die oberste das Dach, und alle dazwischen sind gleichzeitig eine Decke und ein Boden.
- Eine Zone ist ein geschlossener Teil des Gebäudes, mit mindestens einem Boden, einer Decke und Wänden dazwischen.
- Eine Wand, die zu mehr als 2 Zonen gehört, ist eine Trennwand.
- ...

Jede Definition wird als Ruby Methode implementiert.

Es stehen beliebig komplexe Geometrie zur Verfügung, mit denen man diese Algorithmen testen kann. Die obenerwähnten Definitionen funktionieren gut mit "normalen" Gebäuden, würden aber z.B. für Gebäude wie das Bilbao Museum von Frank O. Gehry komplett versagen.

Preprocessing & Simulation (Ruby+Insel)

Nachdem alle notwendige Information aus dem SketchUp Modell extrahiert sind und als Ruby Objekte zur Verfügung stehen, kann ein Simulationsmodell gebaut werden.

Zusätzliche Daten aus anderen Quellen

Es ist nicht möglich, alle notwendigen Daten aus dem SketchUp Modell zu extrahieren, da nicht alle nur von der Geometrie abhängig sind.

- Wetterdaten sind extrem wichtig für die Simulation. Je nachdem welche Simulation erwünscht ist, ist es möglich Normdaten aus Referenzregionen zu benutzen, oder gemessene Daten aus einer Wetterstation.
- Die SketchUp Flächen haben weder eine Struktur noch eine Dicke. Aus diesem Grund müssen extra Parameter eingefügt werden, die Dicke, Dichte, Wärmeleitfähigkeit und Wärmekapazität für jede Wandschicht beschreiben.
- Für jede Zone muss definiert werden ob eine Heizung/Kühlung vorhanden ist, mit welcher Leistung, wann sie an ist und wie sie gesteuert wird.
- Die Fensterflächen müssen definiert werden

Herstellung eines kompletten INSEL Modell

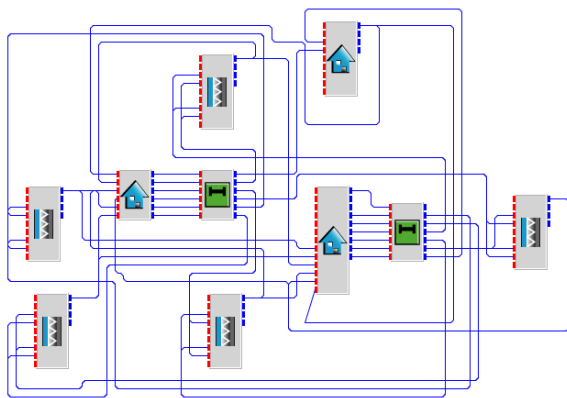


Abbildung 3: INSEL Modell für eine Zone

Da die Geometrie und die Topologie vom Gebäude beliebig komplex sein kann, muss ein INSEL Modell automatisch und dynamisch mit Ruby hergestellt werden. Das Modell besteht aus:

- 1 WALL/WALLX Block pro Wand
- 1 RADI Block pro Zone
- 1 ROOM Block pro Zone
- Notwendige Blöcke für die Zeit, die Heizung und die Wetterdaten
- Jede Zone muss mit jeder ihrer Wände verbunden sein
- Jede Wand muss mit jeder ihrer Zonen verbunden sein
- DELAY Blöcke müssen eingefügt werden gegen algebraischen Schleifen
- WRITE/PLOT/SCREEN Blöcke für die Ergebnisse

Simulation

Nachdem alle Daten zu Verfügung stehen, kann die Gebäudesimulation durchgeführt werden.

Die Verbindung zwischen INSEL & Ruby wurde schon implementiert, so dass man direkt im Ruby Skript die Simulation abrufen kann. Da andererseits Ruby auch mit SketchUp integriert ist, kann man die

INSEL Simulation direkt aus der SketchUp Oberfläche starten.

Die verschiedenen Ergebnisse werden als Ruby Objekte geliefert: die Temperaturverläufe für jede Wand/Zone sind *Arrays* von Zahlen und die Gnuplot Diagramme und Ausgangsdateien werden automatisch in einem Ordner gespeichert.

SketchUp kann diese Information benutzen um z.B. die Farbe von den Zonen zu ändern, je nachdem wie hoch z.B. die maximale Temperatur ist.

Parametrische Analyse können mit *Arrays* einfach definiert werden: Anstatt festzusetzen, dass die Heizung eine Nennleistung von 10kW hat, kann man definieren dass die Nennleistung [5kW,10kW,15kW] sein kann. Es werden automatisch 3 Simulationen durchgeführt, und die Ergebnisse werden in eine Tabelle geschrieben.

Validierung mit VDI 6020

Das entwickelte dynamische Gebäudemodell wurde mit definierten Randbedingungen von dynamischem Raumverhalten aus der VDI 6020 validiert. Exemplarisch ist die Übereinstimmung zwischen Modell und VDI Werten für einen Raum mit hoher Speicherkapazität dargestellt. In diesem ersten Beispiel der VDI wurden 60 Tage simuliert, an denen zwischen 6 und 18 Uhr eine rein konvektive Last eingebracht wird. Zwischen den Raumlufttemperaturen aus der INSEL Simulation und einer TRNSYS Vergleichsrechnung sind keine Unterschiede zu erkennen.

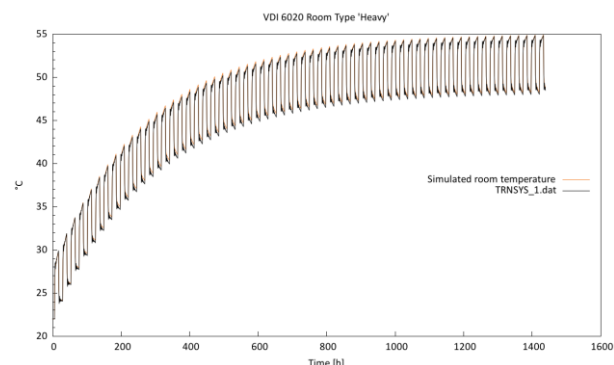


Abbildung 4: Vergleich des modularen dynamischen Gebäudemodells mit TRNSYS Simulationen für das erste Beispiel der VDI 6020.

AUSBLICK

Das Gebäudemodell wird in den nächsten Monaten mit Messdaten eines Mehrfamilienhauses sowie eines Bürogebäudes weiter validiert. Ein weiteres Feature der Sketch-up Modellierung ist die Integration von Verschattungsberechnungen, um solare Einstrahlung auf Fassaden- und Dachflächen genau ermitteln zu können.

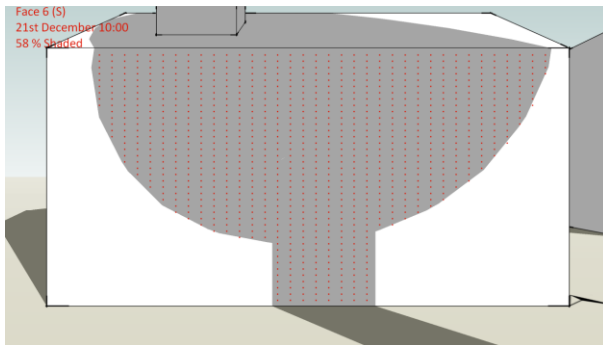


Abbildung 5: Verschattungsbeispiel für eine Gebäudefassade mit Baum

SketchUp kann zwar Verschattungen graphisch darstellen, die Software braucht aber Ruby-Erweiterungen, um die relevanten Informationen extrahieren zu können. Damit können für jedes Bauteil prozentuale Verschattungsanteile für jeden Zeitschritt ermittelt werden (z.B. 58% Verschattung auf einer Südfassade am 21.12. um 10:00).

ZUSAMMENFASSUNG

Am Forschungszentrum zafh.net der Hochschule für Technik wurde ein vollständig modulares Gebäudesimulationsmodell entwickelt, welches automatisiert aus Google Sketch-up Zeichnungen parametrisiert werden kann. Das Modell ermöglicht die dynamische Gebäudesimulation einzelner Objekte, aber auch ganzer Stadtquartiere. Weiterhin sind Verschattungsanalysen integrierbar, um passiv solare Gewinne oder Solarpotentiale zu berechnen. Der Vorteil dieses Ansatzes ist der Zugriff auf alle Berechnungsergebnisse an jeder Stelle des Gebäudes zu jedem Zeitpunkt, die Geschwindigkeit der Lösung durch den Verzicht auf übergeordnete Gleichungslöser und die einfache Erweiterung der Modelle mit erneuerbaren Versorgungssystemen, Netzsimulationen oder ähnliches. Alle entwickelten Komponentenmodelle sind in verschiedenen Simulationssystemen wie INSEL oder Matlab/Simulink voll nutzbar.