Technical Report LBNL-54657

BuildOpt – A New Building Energy Simulation Program that is Built on Smooth Models

Michael Wetter
Simulation Research Group
Building Technologies Department
Environmental Energy Technologies Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720, USA

March 4, 2004

Abstract

Building energy simulation programs compute numerical approximations to physical phenomena that can be modeled by a system of differential algebraic equations (DAE). For a large class of building energy analysis problems, one can prove that the DAE system has a unique once continuously differentiable solution. Consequently, if building simulation programs are built on models that satisfy the smoothness assumptions required to prove existence of a unique smooth solution, and if their numerical solvers allow controlling the approximation error, one can use such programs with Generalized Pattern Search optimization algorithms that adaptively control the precision of the solutions of the DAE system. Those optimization algorithms construct sequences of iterates with stationary accumulation points and have been shown to yield a significant reduction in computation time compared to algorithms that use fixed precision cost function evaluations.

In this paper, we state the required smoothness assumptions and present the theorems that state existence of a unique smooth solution of the DAE system. We present BuildOpt, a detailed thermal and daylighting building energy simulation program. We discuss examples that explain the smoothing techniques used in BuildOpt. We present numerical experiments that compare the computation time for an annual simulation with the smoothing techniques applied to different parts of the models. The experiments show that high precision approximate solutions can only be computed if smooth models are used. This is significant because today's building simulation programs do not use such smoothing techniques and their solvers frequently fail to obtain a numerical solution if the solver tolerances are tight. We also present how BuildOpt's approximate solutions converge to a smooth function as the precision parameter of the numerical solver is tightened.

Disclaimer

1 Introduction

We present BuildOpt, a new multi-zone thermal and daylighting building energy simulation program. BuildOpt is different from existing building energy simulation programs, such as EnergyPlus (Crawley et al., 2001), TRNSYS (Klein et al., 1976) and DOE-2 (Winkelmann et al., 1993), since it is built on models that are defined by differential algebraic equations (DAE system) that are once Lipschitz continuously differentiable in the building design parameters, and since all partial differential equations, ordinary differential equations and algebraic equations are solved simultaneously. The use of smooth models not only allows proving that the DAE system has a unique solution that is once continuously differentiable in the building design parameters, but it is in fact required to achieve convergence of the DAE solver if the solver tolerances are tight. This is a significant observation because today's building energy simulation programs are built on non-smooth models, and their solvers frequently fail to obtain a numerical solution if the solver tolerances are tight.

The use of a DAE solver, as opposed to ad-hoc implemented solvers that are spread throughout the code (which is common in most building energy simulation programs), allows controlling the precision of the numerical approximations to the solution of the DAE system and hence it allows obtaining a function that bounds the approximation error as a function of the solver tolerance. This is required in order for the simulation program to be used with Generalized Pattern Search (GPS) optimization algorithms with adaptive precision cost function evaluations, developed by Polak and Wetter (2003), or by algorithms that are based on the Master Algorithm Model 1.2.36 in Polak (1997). Those optimization algorithms use coarse precision simulations for the early iterations and progressively increase the precision of the simulations. This significantly reduces computation time and allows proving that the optimization algorithm constructs sequences of iterates with stationary accumulation points. To the best of our knowledge, BuildOpt is the first building energy simulation program that can be used to do building design optimizations that provably converge to a stationary point.

Numerical experiments with EnergyPlus and analysis of its source code revealed that it does not seem possible to prove that EnergyPlus computes an approximate solution that converges to a smooth function as the solver tolerances are tightened. In fact, in numerical experiments in which we modeled a building's heating and cooling load and daylighting control, there were about ten solvers that controlled subsystems of the simulation model (such as the heat conduction in the solids, the variable time-step integration of the room air temperature and the initialization of the state variables). We were not able to analyze how the approximation errors of the different solvers were propagated from one model to another, and from one time step to the next, and the code became unstable as we increased the solver tolerances. In Wetter and Wright (2003a), Wetter and Polak (2003) and (Wetter and Wright, 2003b), it is shown that a building's annual energy consumption computed by EnergyPlus is discontinuous in the building design parameters, and that the discontinuities are in some cases on the order of 2% of the cost function value. This caused in some numerical experiments the optimization algorithms to jam. In order to have a building energy simulation program that can be used to perform building design optimizations that provably find a stationary point of the cost function, we developed BuildOpt.

One may ask why we developed our own simulation program rather than having used the IDA-ICE program (Björsell et al., 1999; Sahlin and Bring, 1991), which is an equation-based building energy analysis program that has a large library of simulation models. IDA-ICE generates from equation-based models a DAE system which it solves simultaneously. Discussions with its developer showed that IDA-ICE might indeed be a promising tool for use with our optimization algorithms. However, without extensive numerical experiments and code analysis, it is not possible to conclude that IDA-ICE satisfies our requirements. Furthermore, in case of bad performance of our optimization algorithms, it would have been hard if not impossible to detect why our algorithms did not work as expected. Therefore, for the initial experiments of our optimization algorithms, we preferred to develop our own code.

The paper is structured as follows. First, we present the optimization problem and the assumptions that the simulation program needs to satisfy in order to be used with our GPS algorithms with adaptive precision cost function evaluations. Next, we characterize BuildOpt's physical model, explain some of the models that are implemented in BuildOpt, and explain some of the smoothing techniques that are used in implementing the models. Then, we present numerical experiments that compare how the smoothing techniques affect the convergence of the DAE solver and verify that the numerical approximations converge to a smooth function as precision is increased.

2 Nomenclature

2.1 Conventions

1. Vectors are always column vectors, and their elements are denoted by superscripts.

2. Elements of a set or a sequence are denoted by subscripts.

3. $f(\cdot)$ denotes a function where $(\cdot)$ stands for the undesignated variables. $f(x)$ denotes the value of $f(\cdot)$ for the argument x. $f\colon A \to B$ indicates that the domain of $f(\cdot)$ is in the space A, and that the image of $f(\cdot)$ is in the space B.

4. We say that a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is once (Lipschitz) continuously differentiable if $f(\cdot)$ is defined on $\mathbb{R}^n$, and if $f(\cdot)$ has a (Lipschitz) continuous derivative on $\mathbb{R}^n$.

5. For $\varepsilon, \varepsilon_S \in \mathbb{R}^q_+$, by $\varepsilon \leq \varepsilon_S$, we mean that $0 < \varepsilon^i \leq \varepsilon^i_S$, for all $i = \{1, \ldots, q\}$.

2.2 Symbols

q	dimension of the precision parameter of the numerical solvers
t	time
x	independent parameter in the optimization problem
$a \in A$	a is an element of A
$A \subset B$	A is a subset of B
$\mathbb{R}$	set of real numbers
$\mathbb{R}^q_+$	$\{x \in \mathbb{R}^q \mid x^i > 0,\ i \in \{1,\dots,q\}\}$
$\triangleq$	equal by definition
$\|x\|$	L_2 norm of $x \in \mathbb{R}^n$, defined as $\|x\| \triangleq \left(\sum_{i=1}^n (x^i)^2\right)^{1/2}$

3 Properties of Optimization Problem

3.1 Optimization Problem

We are interested in solving box-constrained problems of the form

$$\min_{x \in \mathbf{X}} f(x), \tag{1}$$

where $\mathbf{X} \triangleq \left\{x \in \mathbb{R}^n \mid l^i \le x^i \le u^i,\ i \in \{1,\dots,n\}\right\}$ is the constraint set, with $-\infty \le l^i < u^i \le \infty$ for all $i \in \{1,\dots,n\}$.

We assume that the cost function is once continuously differentiable and defined as

$$f(x) \triangleq F(z(x,1)), \tag{2}$$

where $F\colon \mathbb{R}^m \to \mathbb{R}$ is once continuously differentiable and $z(x,1) \in \mathbb{R}^m$ is the solution of a semi-explicit nonlinear DAE system with index one (Brenan et al., 1989) of the form

$$\dot{z}(x,t) = h\big(x, z(x,t), \mu\big), \qquad t \in [0,\ 1], \tag{3a}$$

$$z(x,0) = z_0(x), \tag{3b}$$

$$\gamma\big(x, z(x,t), \mu\big) = 0, \tag{3c}$$

where $h\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^m$, $z_0\colon \mathbb{R}^n \to \mathbb{R}^m$ and $\gamma\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$ are once Lipschitz continuously differentiable in all arguments and equation (3c) has, for all $x \in \mathbb{R}^n$ and for all $z(\cdot,\cdot) \in \mathbb{R}^m$, a unique solution $\mu^*(x,z) \in \mathbb{R}^l$ and the matrix with partial derivatives $\partial\gamma(x, z(x,t), \mu^*(x,z))/\partial\mu \in \mathbb{R}^{l \times l}$ is non-singular. The notation $\dot{z}(x,t)$ denotes differentiation with respect to time.

Equation (3) is a DAE system that describes a building energy simulation model after the spatial domains of wall, floor and ceiling constructions have been discretized in a finite number of nodal points. For example, the components of the vector $z(\cdot,\cdot)$ can be the room air temperature, the solid temperature at the nodal points, and the building energy consumption for cooling, heating and lighting, and $\gamma(\cdot,\cdot,\cdot)$ can be a system of nonlinear equations that is used to describe the temperature of elements with negligible thermal capacity, such as window glass.

3.2 Existence of a Unique Smooth Solution of the DAE System

We will now state the assumptions that we use to establish existence, uniqueness and differentiability of the solution $z(\cdot,1)$ of (3).

Assumption 3.1 *With* $\gamma\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$ *as in* (3c)*, we assume that* $\gamma(\cdot,\cdot,\cdot)$ *is once continuously differentiable, and we assume that for all* $x \in \mathbb{R}^n$ *and for all* $z(\cdot,\cdot) \in \mathbb{R}^m$*, equation* (3c) *has a unique solution* $\mu^*(x,z) \in \mathbb{R}^l$ *and that the matrix with partial derivatives* $\partial\gamma(x,z(x,t),\mu^*(x,z))/\partial\mu \in \mathbb{R}^{l\times l}$ *is non-singular.* □

By using the Implicit Function Theorem (Polak, 1997), one can show that Assumption 3.1 implies that the solution of (3c), i.e., the $\mu^*(x,z)$ that satisfies $\gamma\big(x,z(x,t),\mu^*(x,z)\big) = 0$, is unique and once continuously differentiable in x and z. Therefore, to establish existence, uniqueness and differentiability of $z(\cdot,1)$, we can reduce the DAE system (3) to an ordinary differential equation, which will allow us to use standard results from the theory of ordinary differential equations. To do so, we define for $x \in \mathbb{R}^n$, for $t \in [0,\,1]$ and for $z(x,t) \in \mathbb{R}^m$ the function

$$\widetilde{h}(x,z(x,t)) \triangleq h\big(x,z(x,t),\mu^*(x,z)\big), \tag{4}$$

and write the DAE system (3) in the form

$$\dot{z}(x,t) = \widetilde{h}\big(x,z(x,t)\big), \qquad t \in [0,\,1], \tag{5a}$$

$$z(x,0) = z_0(x). \tag{5b}$$

We will use the notation $\widetilde{h}_x(x,z(x,t))$ and $\widetilde{h}_z(x,z(x,t))$ for the partial derivatives $(\partial/\partial x)(\widetilde{h}(x,z(x,t)))$ and $(\partial/\partial z)(\widetilde{h}(x,z(x,t)))$, respectively. We will make the following assumption.

Assumption 3.2 *With* $\widetilde{h}\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ *and* $z_0\colon \mathbb{R}^n \to \mathbb{R}^m$ *as in* (5)*, we assume that*

1. *the initial condition* $z_0(\cdot)$ *is continuously differentiable,*
2. *there exists a constant* $K \in [1,\infty)$ *such that for all* $x',x'' \in \mathbb{R}^n$ *and for all* $z',z'' \in \mathbb{R}^m$*, the following relations hold:*

$$\|\widetilde{h}(x',z') - \widetilde{h}(x'',z'')\| \le K\left(\|x'-x''\| + \|z'-z''\|\right), \tag{6a}$$

$$\|\widetilde{h}_x(x',z') - \widetilde{h}_x(x'',z'')\| \le K\left(\|x'-x''\| + \|z'-z''\|\right), \tag{6b}$$

and

$$\|\widetilde{h}_z(x',z') - \widetilde{h}_z(x'',z'')\| \le K\left(\|x'-x''\| + \|z'-z''\|\right). \tag{6c}$$

□

Now we can use the following theorem, which is a special case of Corollary 5.6.9 in Polak (1997), to show that $f(\cdot) \triangleq F(z(\cdot,1))$ is once continuously differentiable.

Theorem 3.3 *Suppose that* $F\colon \mathbb{R}^m \to \mathbb{R}$ *is once continuously differentiable on bounded sets, that Assumptions 3.1 and 3.2 are satisfied and that* $f\colon \mathbb{R}^n \to \mathbb{R}$ *is defined by* $f(x) \triangleq F(z(x,1))$*. Then,* $f(\cdot)$ *is once continuously differentiable on bounded sets.* □

3.3 Numerical Solutions of the DAE System

We assume that $z(x,t)$ cannot be evaluated exactly, but that it can be approximated numerically by functions $z^*(\varepsilon,x,t)$, where $z^*\colon \mathbb{R}^q_+ \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ and $\varepsilon \in \mathbb{R}^q_+$ is a vector that contains the precision parameters of the DAE solvers. Hence, we denote by $z^*(\varepsilon,x,t)$ the numerical approximation for the solution $z(x,t)$ of (3), as computed by a simulation program with solver precision parameters ε. Thus, for $\varepsilon \in \mathbb{R}^q_+$ and $x \in \mathbb{R}^n$, we define approximating cost functions $f^*(\varepsilon,x) \triangleq F(z^*(\varepsilon,x,1))$ which are, in general, discontinuous in x due to adaptive algorithms in the DAE solver, such as variable time step integration algorithms or Newton-based solvers.

3.4 Requirements on the Exact Cost Function and on the Approximating Cost Functions

We want to use BuildOpt to evaluate the cost function in Generalized Pattern Search (GPS) algorithms with adaptive precision cost function evaluations (Polak and Wetter, 2003). For those algorithms, we need to make the following assumptions on the solution $z(\cdot,1)$ and its numerical approximations $\{z^*(\varepsilon,\cdot,1)\}_{\varepsilon\in\mathbb{R}^q_+}$.

Assumption 3.4

1. *There exists an error bound function* $\varphi\colon \mathbb{R}^q_+ \to \mathbb{R}_+$ *such that for any bounded set* $\mathbf{S} \subset \mathbf{X}$*, there exists an* $\varepsilon_{\mathbf{S}} \in \mathbb{R}^q_+$ *and a scalar* $K_{\mathbf{S}} \in (0,\infty)$ *such that for all* $x \in \mathbf{S}$ *and for all* $\varepsilon \in \mathbb{R}^q_+$*, with* $\varepsilon \le \varepsilon_{\mathbf{S}}$*,*

$$|z^*(\varepsilon,x,1) - z(x,1)| \le K_{\mathbf{S}}\,\varphi(\varepsilon). \tag{7}$$

   Furthermore,

$$\lim_{\|\varepsilon\|\to 0} \varphi(\varepsilon) = 0. \tag{8}$$

2. *The function* $z\colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ *is once continuously differentiable.* □

Note that we allow the functions $\{z^*(\varepsilon,\cdot,1)\}_{\varepsilon\in\mathbb{R}^q_+}$ to be discontinuous. Examples of error bound functions $\varphi(\cdot)$ can be found in Polak and Wetter (2003) and in Wetter and Polak (2003).

4 BuildOpt Simulation Program

Many if not all of today's detailed building simulation programs are built on models that do not satisfy Assumptions 3.1 and 3.2. Furthermore, the numerical experiments presented in Wetter and Wright (2003a), in Wetter and Polak (2003), and in Wetter and Wright (2003b) show that approximating cost functions computed by EnergyPlus can have discontinuities on the order of 2% of the cost function value. In some numerical experiments, this caused optimization algorithms to jam.

To ensure that the simulation program used in the optimizations is built on models that are smooth in the input data and to ensure that controlling the approximation error is possible, we developed BuildOpt, a new building energy simulation program.

4.1 Models

BuildOpt is built on detailed simulation models that consist of 30,000 lines (1.2 MB) of C/C++ code. We will here only give a brief overview of the models to give an impression of the model complexity.

The diffuse solar irradiation is computed using the model of Perez et al. (1990, 1987) and the radiation temperature of a cloudy sky is computed using the model of Martin and Berdahl (1984). To compute the heat conduction in opaque materials, with possibly composite layers, the Galerkin method (Evans, 1998; Strang and Fix, 1973) is used for the spatial discretization. The spatial discretization results in systems of ordinary differential equations. The systems are coupled to other constructions via long-wave radiative heat exchange, and are coupled to the room air temperatures via convective heat transfer. The short-wave radiation through multi-pane windows is computed using a model similar to the one used in the Window 4 program (Finlayson et al., 1993). The daylight illuminance is computed with a model based on view-factors that is similar to the model in the DeLight program of Vartiainen (2000). All equations are solved simultaneously, as explained in Section 4.3.

4.2 Smoothing Techniques

BuildOpt differs from other building simulation programs in that it uses various smoothing techniques to make all model equations as well as the table look-ups (used in Perez' model), hourly schedules of internal heat gains and weather data once Lipschitz continuously differentiable in the state variables, in the model parameters and in time. Smoothing is required to satisfy Assumption 3.2, it significantly reduces the computation time and it is required to achieve convergence of the DAE solver if the solver tolerance is tight.

The building blocks used to formulate once Lipschitz continuously differentiable models are as follows. For $s \in \mathbb{R}$ and for some $\delta > 0$, we defined a once Lipschitz continuously differentiable approximation for the Heaviside function as

$$\widetilde{\mathrm{H}}(s;\delta) \triangleq \begin{cases} 0, & \text{for } s < -\delta, \\ \frac{1}{2}\left(\sin\left(\frac{s}{\delta}\frac{\pi}{2}\right)+1\right), & \text{for } -\delta \leq s < \delta, \\ 1, & \text{for } \delta \leq s. \end{cases} \tag{9}$$

We parametrized (9) by $\delta > 0$ to be able to take the scaling of s into account. Equation (9) is used to define a once Lipschitz continuously differentiable approximation for the max function as

$$\widetilde{\max}(a,b;\delta) \triangleq a + (b-a)\,\widetilde{\mathrm{H}}(b-a;\delta). \tag{10}$$

This function is then, for example, used to smoothen the convective heat transfer coefficient between a wall surface and the room air as we will now explain. A commonly used equation for the convective heat transfer coefficient due to natural convection between a wall surface at temperature T and the room air, which we assume for this explanation to have zero temperature, is $h(T) = 1.310|T|^{1/3}$. Hence, the convective heat transfer per unit area is $q(T) = 1.310|T|^{1/3}T$,

which has a derivative that fails to be Lipschitz continuous near zero. Consequently, we used for the convective heat transfer coefficient the once Lipschitz continuously differentiable approximation $\widetilde{h}(T) = 1.310\ \widetilde{\max}(1,\ |T|^{1/3};0.1)$.

To interpolate the weather data for time instants that do not coincide with time stamps in the weather data file, we used cubic splines. To interpolate values of internal loads for people, lighting and electric equipment, which are specified by hourly schedules, we used the smooth Heaviside function (9) with $\delta = 1/2$ hour.

Thus, BuildOpt's simulation models are written in such a way that the functions $h\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^m$, $z_0\colon \mathbb{R}^n \to \mathbb{R}^m$ and $\gamma\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$, defined in (3), are implemented so that the Assumptions 3.1 and 3.2 are satisfied. We do not believe that there is any other building energy simulation program that is built on models that are as detailed as the models in BuildOpt and that satisfies Assumptions 3.1 and 3.2.

4.3 Solving the Equations

BuildOpt's models are linked with the DAE solver DASPK (Brenan et al., 1989; Brown et al., 1994). The DASPK solver uses a variable time-step, variable order Backward-Differentiation Formula.

To solve the DAE system (3), the DASPK solver requires the simulation model to be written in the residual form

$$G\left(t,\, v(x,t),\, \dot{v}(x,t)\right) = \begin{pmatrix} \dot{z}(x,t) - h(x,\, z(x,t),\, \mu^*(x,z)) \\ \gamma(x,\, z(x,t),\, \mu^*(x,z)) \end{pmatrix} = 0, \tag{11}$$

where $v(x,t) \triangleq (z(x,t), \mu^*(x,z))^T \in \mathbb{R}^{m+l}$ is the vector of differential variables $z(x,t)$ and of algebraic variables $\mu^*(x,z)$, which are the solution of (3c). Given initial values of the differential variables $z(x,0)$, DASPK computes consistent initial conditions $\dot{z}(x,0)$ and $\mu^*(x,z(x,0))$, or conversely, given $\dot{v}(x,0)$, it computes consistent values for $v(x,0)$ (see Brown et al. (1998)).[1] At each time step $t \in [0,\, 1]$, DASPK passes to BuildOpt a $\widehat{t} > t$, a $\widehat{v}(x,\widehat{t})$ and a $\widehat{\dot{v}}(x,\widehat{t})$, where $\widehat{\dot{v}}(x,\widehat{t})$ is approximated by backward differences[2] and BuildOpt returns to DASPK the residual vector $G\left(\widehat{t}, \widehat{v}(x,\widehat{t}), \widehat{\dot{v}}(x,\widehat{t})\right) \in \mathbb{R}^{m+l}$. This process is repeated iteratively until all convergence tests in DASPK are satisfied. Our simulation model is too big to obtain an analytical expression for the iteration matrices $G_v(\cdot,\cdot,\cdot)$ and $G_{\dot{v}}(\cdot,\cdot,\cdot)$ used by DASPK. Hence, we configured DASPK so that it approximates the iteration matrices using finite differences. The linear system of equations that arises in the Newton iterations is solved using a direct method. We note that more efficient solution strategies could be implemented in our code, but we have not yet pursued such improvements. For example, the linear system could be solved using a sparse matrix solver or Krylov iterations.

[1] We say that initial conditions $v(x,0)$ and $\dot{v}(x,0)$ are consistent if $G(0, v(x,0), \dot{v}(x,0)) = 0$.

[2] E.g., if the Implicit Euler method is used, then $\widehat{\dot{v}}(x,\widehat{t})$ is replaced by $(v(x,\widehat{t}) - v(x,\widehat{t}-\delta))/\delta$, where $\delta \in \mathbb{R}$ is the integration time step.

Furthermore, we currently check only in the computationally most expensive models if the input data are the same as in the last model evaluation, in which case no model evaluation is required.[3]

4.4 Model Validation

In Wetter et al. (2004), the thermal simulation model is validated using the ANSI/ASHRAE Standard test procedure 140-2001 (ASHRAE, 2001), and the daylighting simulation is validated using benchmark tests from Laforgue (1997) and Fontoynont et al. (1999), which were produced in the Task 21 of the International Energy Agency (IEA) Solar Heating & Cooling Program. The results of BuildOpt show good agreement with the results of the other validated programs.

5 Numerical Experiments

We will now describe how the smoothing techniques affect the convergence of the DASPK solver and consequently reduce the computation time. For the numerical experiments, we did computations of the annual source energy consumption for heating, cooling and lighting of an office building in Houston, TX. We simulated three representative spaces: a north and a south facing room and a hallway between the rooms. The simulation model is described in Wetter and Wright (2003b). In Tab. 1 we show for different precision parameters ε the normalized number of evaluations of the residual function $G(\cdot,\cdot,\cdot)$, defined in (11), in an annual simulation. For BuildOpt, the number of residual evaluations is proportional to the computation time. A normalized computation time of 1.0 corresponds to 33.4 minutes on one 2.2 GHz AMD processor using the Linux $2.4.18-3$ kernel. The first three columns in Tab. 1 are defined as follows: In the column labeled "model equations", "smooth" means that the smoothing of the model equations is enabled (i.e., all model equations, but not necessarily the hourly schedules, are once Lipschitz continuously differentiable in the state and in time), and "non-smooth" means that the smoothing is disabled. In the column labeled "internal loads", "smooth" means that internal loads due to people, lights, and electric equipment, which are all specified by hourly schedules, are interpolated using the function $\widetilde{H}(\cdot;\delta)$, as defined in (9), with $\delta = 1/2$ hour, "linear" means that we used linear interpolation, where the change from one value to the next occurs over one hour, and "step" means that the hourly schedules are implemented as step functions. In the column labeled "weather data", "cubic" means that we used cubic splines to interpolate the weather data, and "linear" means that we used linear interpolation. For all combinations of these smoothing techniques, we did five annual simulations with solver tolerance settings $\varepsilon \in \{10^{-m}\}_{m=1}^{5}$.

We observed that we were only able to compute high precision approximations when we used smooth models, which will hardly surprise any numerical analyst. The reason is that the DASPK solver uses Taylor expansions to approximate solutions of nonlinear equations and to replace derivatives by finite difference approximations, which is common to any Newton-based solver. However, if the equations being solved are not differentiable, then the Taylor expansions

[3] When DASPK approximates the elements of the iteration matrices $G_v(\cdot,\cdot,\cdot)$ and $G_{\dot{v}}(\cdot,\cdot,\cdot)$, many models are repetitively evaluated with no change in input data.

Smoothing			Solver tolerance ε				
Model equations	Internal loads	Weather data	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
smooth	smooth	cubic	0.011	0.080	0.169	0.497	1
smooth	linear	cubic	0.011	0.078	0.170	0.503	1.048
smooth	step	cubic	0.012	0.091	*	*	*
smooth	smooth	linear	0.005	0.056	0.299	0.879	2.012
smooth	linear	linear	0.006	0.054	0.300	0.889	2.044
smooth	step	linear	0.007	0.069	0.454	*	*
non-smooth	smooth	cubic	0.011	0.078	0.233	*	*
non-smooth	linear	cubic	0.011	0.078	0.238	*	*
non-smooth	step	cubic	0.012	0.093	*	*	*
non-smooth	smooth	linear	0.009	0.093	0.511	*	*
non-smooth	linear	linear	0.009	0.093	0.521	*	*
non-smooth	step	linear	0.010	0.110	*	*	*

Table 1: *Normalized number of calls to $G(\cdot,\cdot,\cdot)$ in an annual simulation with different solver tolerances and different smoothing. An asterisk "*" means that the DAE solver failed to converge in 25 time steps, in which case the simulation stopped.*

are inaccurate or even completely wrong, which can cause the Newton search direction to point away from the solution of the equation. However, in practice we observe that building simulation programs are built on non-smooth models and contain Newton-based solvers that frequently fail to find a solution if the solver tolerances are tight. This is what we also observed in BuildOpt when we disabled the smoothing techniques. Furthermore, when the solver tolerance was tight, BuildOpt's computation time increased by a factor of two when we changed from cubic splines to linear interpolation of the weather data. This is interesting because many if not all of today's building simulation programs use linear interpolation rather than cubic splines. Thus, we believe that the convergence properties of today's building energy simulation programs could be significantly improved if they were built on smooth models and if they used weather data interpolations that are smooth in time. Numerical solvers that detect state events, such as a change in model equations for some domain of the model input data, are likely to be more robust than DASPK if non-smooth models are used, but the detection of state events is computationally expensive and hence it may be better to prevent state events where possible.

We will now show how the approximate numerical solutions converge to a smooth function as the tolerance of the DASPK solver is tightened.

Let $z^*(\varepsilon, x, 1) \in \mathbb{R}$ denote the numerical approximation of the annual source energy consumption for heating, cooling and lighting of the office building used in the above numerical experiments, computed by BuildOpt with solver tolerance $\varepsilon \in \mathbb{R}_+$. Here, $x \in \mathbb{R}$ denotes the normalized setpoint for the shading device of the south facing window. Let $\delta(\varepsilon, x)$ denote the relative change

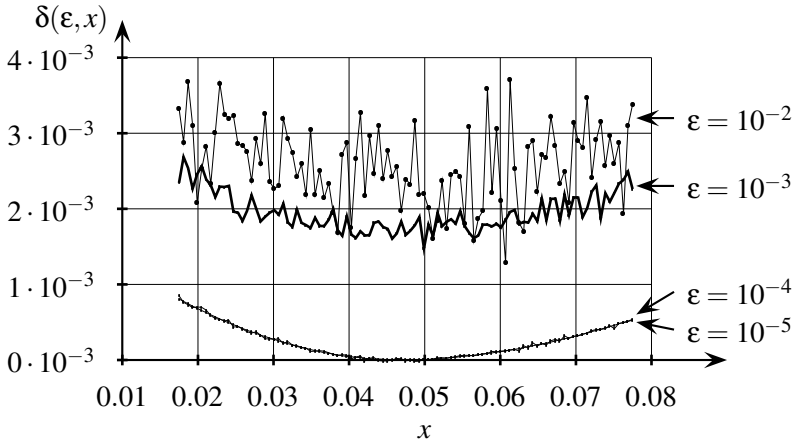

Figure 1: *Relative change of the annual source energy consumption* $\delta(\varepsilon,x)$*, defined in* (12)*, for different precision parameters* ε*. For better visibility of the data series, the support points are connected by lines. The figure is reproduced from Wetter and Polak (2004).*

in the source energy consumption, defined as

$$\delta(\varepsilon,x) \triangleq \frac{z^*(\varepsilon,x,1) - z^*(10^{-5},0.048,1)}{z^*(10^{-5},0.048,1)}. \tag{12}$$

In (12), the argument 0.048 corresponds to the normalized shading control setpoint that yields lowest annual source energy consumption. In Fig. 1, we plot $\delta(\varepsilon,x)$ for different values of ε and x. The figure shows how $z^*(\varepsilon,\cdot,1)$ converges to a smooth function as $\varepsilon \to 0$. The difference between $\delta(10^{-4},\cdot)$ and $\delta(10^{-5},\cdot)$ is almost invisible.

6 Conclusion

Building energy simulation programs can be written so that they compute approximate solutions to a DAE system that converge to a smooth function as the solver tolerance is tightened. This is required if the simulation program is used with GPS algorithms with adaptive precision cost function evaluations which provably construct sequences of iterates with stationary accumulation points.

To obtain convergence of the DAE solver at tight solver tolerance, and to reduce the computation time, it is essential that model equations, the hourly schedules and the weather data are smooth in the state variables and in time. This observation is significant because today's building energy simulation programs are built on non-smooth models and their solvers are known to frequently fail to converge to a solution if the solver tolerances are tight.

7 Acknowledgments

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of the Building Technologies Program of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098.

References

ASHRAE (2001). ANSI/ASHRAE Standard 140-2001, Standard method of test for the evaluation of building energy analysis computer programs.

Björsell, N., Bring, A., Eriksson, L., Grozman, P., Lindgren, M., Sahlin, P., Shapovalov, A., and Vuolle, M. (1999). IDA indoor climate and energy. In Nakahara, N., Yoshida, H., Udagawa, M., and Hensen, J., editors, *Proc. of the 6-th IBPSA Conference*, pages 1035–1042, Kyoto, Japan.

Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1989). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland.

Brown, P. N., Hindmarsh, A. C., and Petzold, L. R. (1994). Using Krylov methods in the solution of large-scale differential-algebraic systems. *SIAM Journal on Scientific Computing*, 15:1467–1488.

Brown, P. N., Hindmarsh, A. C., and Petzold, L. R. (1998). Consistent initial condition calculation for differential-algebraic systems. *SIAM Journal on Scientific Computing*, 19(5):1495–1512.

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., Buhl, W. F., Huang, Y. J., Pedersen, C. O., Strand, R. K., Liesen, R. J., Fisher, D. E., Witte, M. J., and Glazer, J. (2001). EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):443–457.

Evans, L. C. (1998). *Partial differential equations*. American Mathematical Society.

Finlayson, E. U., Arasteh, D. K., Huizenga, C., Rubin, M. D., and Reilly, M. S. (1993). WINDOW 4.0: Documentation of calculation procedures. Technical Report LBL-33943, Lawrence Berkeley National Laboratory, Berkeley CA, USA.

Fontoynont, M., Laforgue, P., Mitanchey, R., Aizlewood, M., Butt, J., Carroll, W., Hitchcock, R., Erhorn, H., Boer, J. D., Dirksmöller, M., Michel, L., Paule, B., Scartezzini, J. L., Bodart, M., and Roy, G. (1999). IEA SHC Task 21/ECBCS Annex 29, Daylight in buildings, Subtask C1: Validation of daylighting simulation programmes. Technical Report T21/C1-/FRA/99-11, International Energy Agency.

Klein, S. A., Duffie, J. A., and Beckman, W. A. (1976). TRNSYS – A transient simulation program. *ASHRAE Transactions*, 82(1):623–633.

Laforgue, P. (1997). IEA SHC Task 21/ECBCS Annex 29, Daylight in buildings, Subtask C1: Draft report of Genelux simulations and other software results. Technical Report T21/C1/97-10, International Energy Agency.

Martin, M. and Berdahl, P. (1984). Characteristics of infrared sky radiation in the United States. *Solar Energy*, 33:321–336.

Perez, R., Ineichen, P., Seals, R., Michalsky, J., and Stewart, R. (1990). Modeling daylight availability and irradiance components from direct and global irradiance. *Solar Energy*, 44(5):271–289.

Perez, R., Seals, R., Ineichen, P., Stewart, R., and Menicucci, D. (1987). A new simplified version of the Perez diffuse irradiance model for tilted surfaces. *Solar Energy*, 39(3):221–231.

Polak, E. (1997). *Optimization, Algorithms and Consistent Approximations*, volume 124 of *Applied Mathematical Sciences*. Springer Verlag.

Polak, E. and Wetter, M. (2003). Generalized pattern search algorithms with adaptive precision function evaluations. Technical Report LBNL-52629, Lawrence Berkeley National Laboratory, Berkeley, CA.

Sahlin, P. and Bring, A. (1991). IDA solver – A tool for building and energy systems simulation. In Clarke, J. A., Mitchell, J. W., and de Perre, R. C. V., editors, *Proc. of the IBPSA Conference*, Nice, France.

Strang, G. and Fix, G. J. (1973). *An Analysis of the Finite Element Method*. Prentice-Hall, Inc.

Vartiainen, E. (2000). Daylight modelling with the simulation tool DeLight. Technical Report TKK-F-A799, Helsinki University of Technology, Finland, Dept. of Engineering Physics and Mathematics.

Wetter, M. and Polak, E. (2003). A convergent optimization method using pattern search algorithms with adaptive precision simulation. In Augenbroe, G. and Hensen, J., editors, *Proc. of the 8-th IBPSA Conference*, volume III, pages 1393–1400, Eindhoven, NL.

Wetter, M. and Polak, E. (2004). Building design optimization using a convergent pattern search algorithm with adaptive precision simulations. Submitted to *Energy and Buildings*.

Wetter, M., Polak, E., and Carey, V. P. (2004). BuildOpt 1.0.1 validation. Technical Report LBNL-54658, Lawrence Berkeley National Laboratory, Berkeley, CA, USA.

Wetter, M. and Wright, J. (2003a). Comparison of a generalized pattern search and a genetic algorithm optimization method. In Augenbroe, G. and Hensen, J., editors, *Proc. of the 8-th IBPSA Conference*, volume III, pages 1401–1408, Eindhoven, NL.

Wetter, M. and Wright, J. (2003b). A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. To appear in: *Building and Environment*.

Winkelmann, F. C., Birsdall, B. E., Buhl, W. F., Ellington, K. L., Erdem, A. E., Hirsch, J. J., and Gates, S. (1993). DOE-2 supplement, version 2.1E. Technical Report LBL-34947, Lawrence Berkeley National Laboratory, Berkeley, CA, USA.