



Efficient and robust optimization for building energy simulation



Shokouh Pourarian^{a,*}, Anthony Kearsley^b, Jin Wen^c, Amanda Pertzborn^d

^a Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA, United States

^b Mathematical and Computational Science Division at the National Institute of Standards and Technology (NIST), Gaithersburg, MD, United States

^c Department of Civil, Architectural and Environmental Engineering, Drexel University, Philadelphia, PA, United States

^d Mechanical Systems and Controls Group of the Energy and Environment Division (EED) of the Engineering Laboratory (EL) at the National Institute of Standards and Technology (NIST), Gaithersburg, MD, United States

ARTICLE INFO

Article history:

Received 26 August 2015

Received in revised form 30 March 2016

Accepted 8 April 2016

Available online 11 April 2016

Keywords:

Building energy systems

HVAC simulation

Numerical method

Efficiency

Robustness

Powell's Hybrid method

Levenberg-Marquardt method

ABSTRACT

Efficiently, robustly and accurately solving large sets of structured, non-linear algebraic and differential equations is one of the most computationally expensive steps in the dynamic simulation of building energy systems. Here, the efficiency, robustness and accuracy of two commonly employed solution methods are compared. The comparison is conducted using the HVACSIM+ software package, a component based building system simulation tool. The HVACSIM+ software presently employs Powell's Hybrid method to solve systems of nonlinear algebraic equations that model the dynamics of energy states and interactions within buildings. It is shown here that the Powell's method does not always converge to a solution. Since a myriad of other numerical methods are available, the question arises as to which method is most appropriate for building energy simulation. This paper finds considerable computational benefits result from replacing the Powell's Hybrid method solver in HVACSIM+ with a solver more appropriate for the challenges particular to numerical simulations of buildings. Evidence is provided that a variant of the Levenberg-Marquardt solver has superior accuracy and robustness compared to the Powell's Hybrid method presently used in HVACSIM+.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In the field of building engineering, software has proven essential for analyzing complex nonlinear thermodynamic interactions between building shells, the surrounding environment, HVAC (heating, ventilating, and air-conditioning) systems and building control strategies. These software packages employ mathematical models of real systems, so a common challenge encountered by software developers is selecting numerical solution methods appropriate for the mathematical structures inherent to the dynamic modeling of energy in buildings. The goal is an accurate, stable and globally convergent solution at each step of the time sequence being simulated.

A wide range of building energy simulation tools has been developed during the past four decades for different purposes [1,2]. They differ in complexity and application purpose ranging from simplified spreadsheet programs to fixed schematic hourly simulation programs, from modular variable time-step simulation programs to specialized simulation programs [3,4]. The evolution of using

analytical solution as well as simplifying assumptions to numerical solution considering the real building dynamic is observable in various generations of building energy simulation tools [4]. The current tools can capture reality much better than earlier tools, but are more complex to use. Currently, publically available simulation tools used to simulate dynamic behavior of building and HVAC system as a fully integrated model include SPARK, Energy-Plus, MODELICA, TRNSYS, HVACSIM+, etc. They employ different solution techniques to solve the equations resulting from mathematical modeling of the constitutive components. In the following a brief review of the solution techniques employed in the above mentioned simulation tools is presented.

1.1. SPARK (Simulation problem analysis and research kernel (SPARK))

SPARK [5], which is similar to a general differential/algebraic equation solver, is an object-oriented software system that can be used to simulate physical systems described in differential and algebraic equations. In order to model any system in SPARK, it should be broken down to the constitutive components or objects and they need to be connected through inputs and outputs. The numerical solution techniques for differential equations

* Corresponding author.

E-mail addresses: SP828@drexel.edu, shpourarian@gmail.com (S. Pourarian).

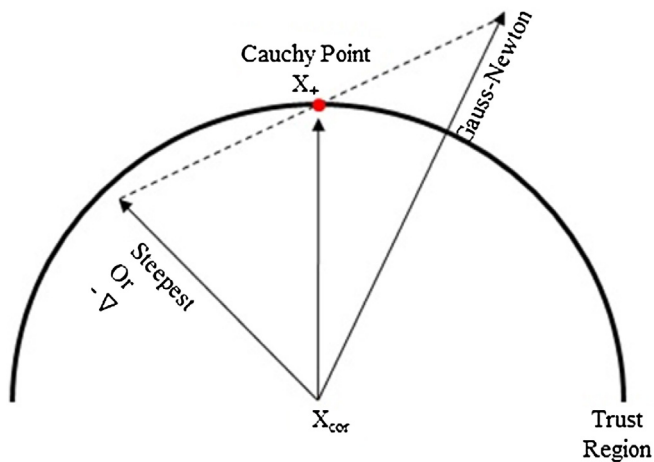


Fig. 1. Visualization of the Cauchy Point for a system of nonlinear equations with trust-region.

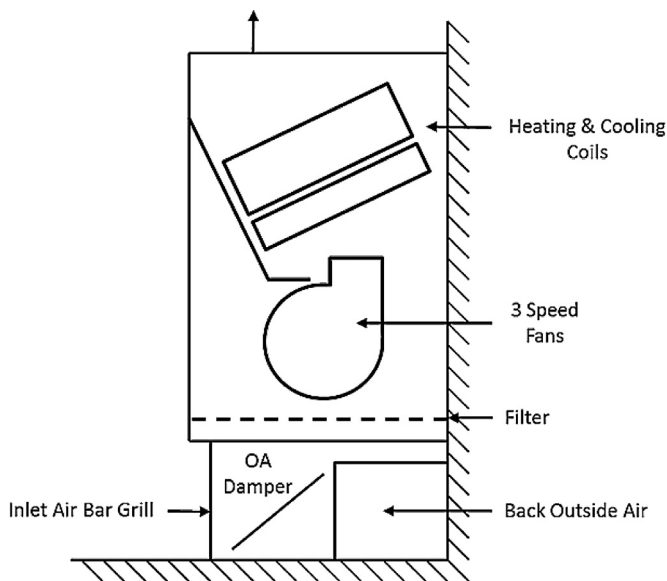


Fig. 2. Fan coil unit configuration.

include: the Euler explicit and implicit methods, the Backward-Forward difference method, the 4th-Order Backward-Forward difference method, the Adams-Bashforth-Moulton method, the PC Euler method and the PC Trapezoidal method. The system of algebraic equations is decomposed into smaller set of equations using graph-theoretic methods to reduce the number of equations to be solved simultaneously. The default solution method of solving the system of algebraic equations is Newton-Raphson. If convergence failure occurs by this method other available numerical methods for the user are Perturbed Newton and Secant method.

1.2. EnergyPlus

EnergyPlus [6] is a building energy analysis and thermal load calculation program which is the combination of the BLAST (Building Loads Analysis and System Thermodynamics) and DOE-2. Given information about the geographic location, physical makeup and mechanical systems of a building, EnergyPlus employ energy and moisture balance equations to calculate the heating and cooling loads that will be required to maintain thermal set points, as well as secondary HVAC system loads, and primary plant energy consumption.

EnergyPlus provides two different numerical integration schemes including 3rd order backward difference, Euler method and one analytical solution algorithm to solve the energy and moisture balance equations. The numerical schemes entail the error propagation into the final solution specially in the long run simulation due to the truncation error while the analytical solution algorithm results in accurate solution without truncation error. [7].

1.3. MODELICA

MODELICA [8], is an object-oriented, equation based language released by Lawrence Berkeley National laboratory (LBNL) which is capable of modeling various complex physical systems in different fields. It also has an open source building library for building energy analysis and control simulation. The library composed of different components give the user the ability of developing the model with no concern about the simulation or numerical solution to the resultant equations of the modeling process. The sets of governing equations obtained through connecting the components are classified as three different kinds [9]: ODEs-Ordinary differential equations for continues-time problems, DAEs- differential algebraic equations for continuous-time problems, Hybrid DAEs-Hybrid differential algebraic equations for mixed continues-discrete problems.

Since the focus of this study is numerical techniques to solve the system of nonlinear algebraic equation and modeling continues-time problems, thus a short review of the methods to solve DAEs is presented. When the simulation problem is a DAE the method of Petzold is used to solve differential algebraic system and some well-known methods for solving ODEs in MODELICA are: The explicit and implicit Euler methods, Multistep methods, The Runge-Kutta methods, The Adams-Bashforth methods.

1.4. TRNSYS (the transient systems simulation)

TRNSYS is a component-based simulation tool to simulate the dynamic and transient performance of the buildings and thermal energy systems which was developed by the Solar Energy Laboratory at the University of Wisconsin Madison since 1975 [10]. The procedure of modeling in TRNSYS is similar to HVACSIM+ which is presented in the following. In TRNSYS, three numerical integration algorithms are used to solve the differential equations including: modified-Euler method (a 2nd order Runge-Kutta method), non-self-starting Heun's method (a 2nd order Predictor-Corrector method) and fourth-order Adams method (a 4th order Predictor-Corrector method).

The employed methods to solve the system of algebraic equations are: the "successive substitution" method and "Powell's" method. In order to solve the system of nonlinear equations TRNSYS uses successive substitution as a default approach and if it is not able to find the converged solution the next option is Powell's method. It is worth mentioning that the default numerical scheme is efficient and reliable for nearly-linear algebraic equations. But for nonlinear equations it hardly can find the solution [10].

The Euler and Runge-Kutta methods are common methods in aforementioned simulation tools. A brief description is provided here on these methods and the interested readers are referred to [11] for a detailed explanation. Euler method is the most basic explicit method to integrate ordinary differential equations (ODE) numerically. In this method, the nonlinear terms of the Tylor series expansion of the function under consideration around the initial point are dropped. Euler's method is referred to as a first order method as the global truncation error is approximately proportional to the time step. Runge Kutta (RK) method is a more accurate method in approximating the solution for ODEs numerically. Because the approximation of the function is accomplished

using slops not only at the beginning of the time step but also half way through it. Depending on the number of slop approximations at each time step the order of RK method is determined. The higher the order of RK method; the higher the accuracy of ODE numerical solution.

1.5. HVACSIM+

The HVACSIM+ dynamic simulation software package [12] developed at the U.S. National Institute of Standards and Technology (NIST) is a component based modeling package. MODSIM as the essence and core program of HVACSIM+ is in charge of solving a system of simultaneous non-linear algebraic equations and integrating differential equations [13].

Each unit in the simulation model is an individual instance of a generic component model representing a specific piece of equipment, envelope element, or control device. Each unit calls its corresponding component model using a serialized “TYPE” (written in all caps, to distinguish from the common use of the word). A TYPE is comprised of one or more nonlinear differential or nonlinear algebraic equations as a FORTRAN 90 routine expressing the component dynamics. Instantiating a TYPE as a unit in the simulation model requires the HVACSIM+ user to link its input and output quantities to those of other, functionally-related units, which contributes equations to the overall system-level set of simultaneous equations that must be solved. Closely-coupled units are first grouped by the user into blocks for simultaneous solution. Blocks are then similarly grouped into a superblock for simultaneous solution. Superblocks are weakly coupled through the state variables and the solver treats each superblock as an independent subsystem of the overall simulation [14]. In each superblock, time evolution and internal solutions are propagated independently of other superblocks. The method used in MODSIM to solve the resulting sets of nonlinear algebraic and differential equations to determine the system state at each time step is based on Powell’s Hybrid (PH) method [15].

The study of solution techniques employed in the building performance tools shows that TRNSYS and HVACSIM+ are using similar approaches to solve system of nonlinear algebraic equations. The authors, together with some earlier literature (ASHRAE-RP 1312 [16], ASHRAE-RP 825 [17]) have indicated that the Powell’s method could fail to converge to a solution, especially when used in the HVACSIM+ environment. Thus, the goal of this study is to propose an alternative to this method and to compare the alternative with the Powell’s method.

In this paper, a numerical study is designed to compare two solution techniques: Powell’s Hybrid (PH) and Levenberg-Marquardt (LM) techniques in terms of their robustness and accuracy. In this numerical study, a dynamic model is developed in HVACSIM+ for a fan coil unit (FCU) interacting with a building zone that the FCU serves. The model is composed of a system of nonlinear algebraic and ordinary differential equations to be solved at each time step in a sequence. In this study, the PH algorithm, as implemented in SNSQ, is replaced by the LM algorithm to solve the identical problem. What follows is a numerical comparison between these two methods when each is employed as the primary solver. This was accomplished by keeping the HVACSIM+ architecture and structure identical and substituting between the PH algorithm and the LM algorithm.

The use of a common platform and task to compare performance of these two methods has a precedent in Shterenlikht et al. in the field of plasticity modeling [18], where the task involved integrity assessment and ductile fracture simulation in simple to complex models of structures. They found the LM method outperformed PH in robustness and accuracy. It is desired to study whether similar conclusions can be drawn when these two techniques are applied in the HVACSIM+ environment to simulate dynamic building systems.

Table 1
Levenberg Marquardt method algorithm.

Input: A function $F: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ and initial estimate $x_0 \in \mathfrak{R}^n$ and output parameters b
Output: A vector $x^* \in \mathfrak{R}^n$ where $F(x^*) \approx b$

```

1  $k := 0; \eta := 2; x := x_0$ 
2  $A := J^T J; r_x := b - F(x); g := J^T r_x$ 
3 converge  $:= (\|g\| \leq \epsilon_g); \mu := \tau \max_i A_{ii}$ 
4 while  $(k < k_{max} \text{ and } \text{converge} = \text{false})$  do
5    $k = k + 1$ 
6   Repeat
7     Solve  $(A + \mu I)\delta_x = g$ 
8     if  $(\|\delta_x\| \leq \epsilon_x \|x\|)$  then
9       converge  $:= \text{true}$ 
10    else
11       $x_+ = x + \delta_x; \rho = \frac{(\|r_x\|^2 - \|b - F(x_+)\|^2)}{(\delta)^T (\mu(\delta) + g)}$ 
12      if  $\rho > 0$  then
13         $x = x_+$ 
14         $A = J^T J; r_x = b - F(x); g = J^T r_x; \text{converge} = (\|g\| \leq \epsilon_g)$ 
15         $\mu = \mu * \max(1/3, 1 - (2\rho - 1)^3); v = 2$ 
16      else
17         $\mu = \mu * v; v = 2v$ 
18  Until  $(\rho > 0)$  or stop  $x_+ = x$ 

```

In this paper, Section 2 summarizes the LM and PH methods while Section 3 describes the numerical study and implementation of both methods in HVACSIM+. The numerical results and comparison are also presented in this section. Finally, Section 4 provides conclusions and future research directions.

2. Simulation description

The subroutine SNSQ in MODSIM implements PH, which seeks to find a zero of a system of N nonlinear, continuously differentiable functions in N variables. The N variables are representative of state variables defined in the model and the N functions are the governing equations in the physical units. The system of nonlinear algebraic equations can be rewritten more generally in vector form as

$$F(x) = 0 \quad (1)$$

where $F: \mathfrak{R}^N \rightarrow \mathfrak{R}^N$, $x \in \mathfrak{R}^N$ is the vector function of primal variables. What follows is a short description of both the LM method and the PH method. The interested reader is referred to more thorough references on both the LM method [19,20] and the PH method [21,22].

2.1. Levenberg Marquardt method

In this section we summarize the LM algorithm; Table 1 presents pseudo-code for the method as implemented. The interested reader is referred to [19,20] for a more detailed description. The LM method has become a standard method for solving systems like Eq. (1). Loosely speaking, LM can be thought of as a combination of both steepest descent and the Gauss-Newton methods in so far as the algorithm behaves like steepest descent when iterates are far from a local minimizer and when iterates draw closer to a local minimizer the method becomes the Gauss-Newton method. Making clear the distinction between being ‘close’ and ‘far’ from a local solution can be made more mathematically justified (see, for example Ref. [23,24]). To help with a comparison, a short description of LM is presented here. In this implementation of LM, specific norms $\|\cdot\|$ are not chosen because in this case, all norms are equivalent; the interested reader is referred to more extensive treatments presented in [25]. The LM method is based on a linear approximation to F in a neighborhood of the point x . The Jacobian matrix J , which is com-

prised of the partial derivatives of F , $\partial F/\partial x$ can be approximated by a Taylor series with step size δ_x ,

$$F(x + \delta_x) \approx F(x) + \frac{\partial F}{\partial x} \delta_x = F(x) + J\delta_x \quad (2)$$

and forms the basis of the iterative technique. In this way, the initial approximation x_0 produces iterates x_i , which seek to converge to the local minimizer x_* . Based on this formulation, the goal at each step is to minimize the residual,

$$\|b - F(x + \delta_x)\| \approx \|b - F(x) - J\delta_x\| = \|b - J\delta_x\| \quad (3)$$

where b is defined by,

$$F(x_*) \approx b \quad (4)$$

In this case the step δ_x is a solution to a linear least squares problem and the solution occurs when $J\delta_x - r_x$ is orthogonal to the column space of J leading to the observation that $J^T(J\delta_x - r_x) = 0$. The Gauss-Newton step, δ_{gn} , solves the so-called normal equations,

$$J^T J \delta_{gn} = J^T (b - F(x)) = J^T r_x \quad (5)$$

when higher order terms are neglected the matrix $J^T J$ approximates the Hessian matrix $r_x^T r_x$. In this case $J^T r_x$ is along the steepest descent direction as $-J^T r_x$ is the gradient of $\frac{1}{2} r_x^T r_x$. In practice the augmented normal equation is:

$$(J^T J + \mu I) \delta_x = J^T r_x \text{ where } \mu > 0 \quad (6)$$

here, I is the appropriately sized identity matrix. The process of adjusting μ is regulated by monitoring the updates in primal variables $x + \delta_x$. If this update results in a decrease in the squared residuals $r_x^T r_x$ then the change that resulted from the μ is accepted and the algorithm continues to a new iteration. If not, the term μ can be increased and the system is solved again until a value of μ results in a sufficient decrease in the squared residuals $r_x^T r_x$. As Table 1 demonstrates, this adjustment takes place at every iteration of the LM algorithm. The larger the μ the more diagonally dominant the normal equations become and the closer the solution to these equations draw to the steepest descent, $J^T \delta_x$. The magnitude of the residual also decreases significantly with increasing μ . Clearly the smaller the steps the greater the robustness, but slower convergence follows. This regularization is also employed for rank deficient matrices J as it ensures the linear system being solved is always positive definite for positive μ . This method terminates, in most implementations, when at least one of the following conditions holds:

- The maximum number of iterations is reached,
- The norm of the gradient is sufficiently small,
- The norm of the step is sufficiently small.

2.2. Powell's Hybrid method

Similar to the LM method, the Powell's Hybrid method seeks to combine both Gauss-Newton and steepest descent, however, in this case the convergence is controlled through a trust-region. Trust-region methods have become a mainstay of nonlinear optimization strategies and are used in a wide variety of applications. Employing a trust-region formulation, the objective function F is used to construct a quadratic model function, H , so that in a neighborhood of a current iterate about which H is constructed, the functions H and F are similar. The function H is said to be trusted to accurately model F in a region with weighted radius Π that is centered at the current iterate. Using this strategy, a candidate step can be calculated by minimizing H over the trust-region. The model function is:

$$H(\delta) = \left(r_x^T r_x - 2(J^T r_x)^T \delta + \delta^T J^T J \delta \right) \quad (7)$$

Table 2
Powell's Hybrid method algorithm.

| | |
|----------------|---|
| Input: | A function $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and initial estimate $x_0 \in \mathbb{R}^n$ and output parameters b |
| Output: | A vector $x^* \in \mathbb{R}^n$ where $F(x^*) \approx b$ |
| 1 | $k := 0; \Pi := \Pi_0; x := x_0$ |
| 2 | $A := J^T J; r_x := b - F(x); g := J^T r_x$ |
| 3 | converge $:= (\ g\ \leq \epsilon_g)$ |
| 4 | while ($k < k_{max}$ and converge = false) do |
| 5 | $k = k + 1$ |
| 6 | $\delta_{sd} = \frac{\ g\ ^2}{\ Jg\ } g$ |
| 7 | Gauss Newton = false |
| 8 | Repeat |
| 9 | if ($\ \delta_{sd}\ \geq \Pi$) then |
| 10 | $\delta_d := \frac{\Pi}{\delta_{sd}} \delta_{sd}$ |
| 11 | else |
| 12 | if not Gauss Newton then |
| 13 | Solve $A \delta_{gn} = g$ |
| 14 | Gauss Newton = true |
| 15 | if ($\ \delta_{sd}\ \leq \Pi$) then |
| 16 | $\delta_d = \delta_{gn}$ |
| 17 | else |
| 18 | $\delta_d = \delta_{sd} + \alpha(\delta_{gn} - \delta_{sd})$ |
| 19 | Choose α so that $\ \delta_d\ = \Pi$ |
| 20 | if ($\ \delta_d\ \leq \epsilon_\delta$) then |
| 21 | Converge = true |
| 22 | else |
| 23 | $x_+ = x + \delta_d; \rho = \frac{(\ r_x\ ^2 - \ b - F(x_+)\ ^2)}{H(0) - H(\delta_d)}$ |
| 24 | if ($\rho > 0$) then |
| 25 | $x = x_+$ |
| 26 | $A := J^T J; r_x := b - F(x); g := J^T r_x$ |
| 27 | Converge $:= (\ g\ \leq \epsilon_g)$ |
| 28 | Update Π converge $:= (\Pi \leq \epsilon_x \ x\)$ |
| 29 | Until $\rho > 0$ or converge |

and a candidate step can be attained by solving:

$$\min_{\delta} H(\delta) \text{ subject to } \|\delta\| \leq \Pi \quad (8)$$

In practice, the trust region radius is selected based on the ability of the model to 'fit' the objective function – strong agreement between the approximation and the original model suggest a strong value of Π . The solution to the trust-region in Eq. (1) can be seen in Fig. 1. In his seminal paper, Powell [22] used piecewise linear trajectories consisting of two line segments in order to approximate the solution curve. In other words, rather than solving the normal equations directly, which can be expensive, Powell approximated the solution using two less expensive calculations. The first segment emanates from the current approximation to the so-called *Cauchy point* which is the unconstrained minimum of the objective function along the steepest descent (or gradient) plane, $J^T r_x$ and is given by:

$$\delta_{sd} := \frac{g^T g}{g^T J^T J g} g \quad (9)$$

the second component runs from this δ_{sd} to the Gauss-Newton step. Table 2 summarizes the Powell's Hybrid algorithm.

3. Numerical case study

The robust convergence capability and application of the LM and PH methods in the building energy simulation tool HVACSIM+ is examined in this numerical study. For this purpose a fan coil unit, a common secondary HVAC system, is modeled. The FCU model created in HVACSIM+ [26] underwent simulation with both methods by keeping the model structure and architecture identical and substituting between the PH and LM algorithms. What follows are the physical and geometrical characteristics of the FCU under study and the control strategy of FCU operation. The specific FCU studied

Table 3
Category and number of state variables.

| Category | Pressure | Flow rate | Temperature | Control | Other (Fan rotational speed) | Power | Humidity |
|---------------------------|----------|-----------|-------------|---------|------------------------------|-------|----------|
| Number of state variables | 7 | 8 | 16 | 14 | 1 | 8 | 4 |

in this paper is a vertical floor mounted four pipe hydronic system including three parallel fans run by two electric motors with three speeds: high, medium and low. The FCU modulates the amount of ventilation supplied to its zone by using a motorized damper in the outside air connection at the back of the unit. Fig. 2 shows the configuration of the FCU with its components and their arrangements.

To simulate the FCU interacting with the corresponding zone in HVACSIM+, the components (units) models are grouped into five blocks and each block constitutes a superblock (i.e., equations of which are solved simultaneously). The final model consists of five superblocks:

(1) control logic (SB1), (2) actuators (SB2), (3) fluid (i.e., mass flow and pressure, SB3), (4) thermal (temperature and humidity, SB4), and (5) sensors (SB5). Fig. 3 illustrates this HVACSIM+ structure for the FCU model. As illustrated in Fig. 3, there are five superblocks, represented by SB1 to SB5. Each rectangular box represents a unit (described by its title). The TYPE number used to model each unit is also provided, such as T479 (TYPE 479) for fan coil unit supply air temperature control. C represents control signal, N represents rotational speed and T represents temperature. For example, it can be seen that the mixed air and room air temperature signals (C11 and C10), sent from the sensors superblock (SB5) to the controls superblock (SB1) result in positioning of the heating and cooling coil valves and the mixed air damper (SB2). Simultaneous solution of mass-pressure equations occurs in the fluid superblock (SB3), while energy balance equations are solved simultaneously in the thermal superblock (SB4). The superblocks are weakly coupled to each other. The solution of each superblock (obtained to some specified convergence tolerance) at time step n is used in other superblock that are coupled at time step $n + 1$. For example, the temperatures from SB4 at time step n are used to determine flow properties in SB3 at time step $n + 1$. The simulation runs for each method with a minimum time step of 2 s and a maximum time step of 10 s to confine the solver as it automatically adjusts the time step and avoid inaccuracy and instability of the simulation. The model of the FCU interacting with two exterior building zones, east and south facing, is simulated on a summer test day. These zones are referred to as rooms in the following discussion. In summer, the outdoor air damper is fully closed and the fan speed is normally set at high. When the FCU is operating properly, the controller compares room temperature to the cooling set-point (74F (23.33C)) and heating set-point (70F (21.11C)). If the actual room temperature is greater than (cooling set-point $-1F$ (0.56C)), the FCU is in cooling mode and if it is less than (heating set-point $+1F$ (0.56C)), the FCU is in heating mode. A dedicated proportional integral derivative (PID) loop is enabled for each mode to control the cooling or heating valve position. A PID loop is a means of regulating a process quantity (room temperature) by compensating it with closed-loop feedback of its error (difference between the room temperature and set-point), with the compensation amount computed linearly using three gain coefficients.

As long as room temperature lies between the heating and cooling set-points, the room is considered to be comfortable and the control system will not adjust the valve and fan settings in the FCU unit. The room air temperature signal is passed from the sensors superblock to the controls superblock, which calculates the required position of the heating and cooling coil valves. Valve positions as determined in the control superblock are passed to other superblocks as appropriate. Simultaneous solution of mass-pressure equations occurs in the air flow superblock, while

energy balance equations are solved simultaneously in the thermal superblock. As shown in Table 3, the total number of variables in the FCU simulation is 58. In this table the category and number of variables in those categories are listed. The Iowa Energy Center Energy Resource Station (ERS) provides weather condition and experimental data necessary for simulation and validation of the FCU model. Additional details of this model, including a discussion of model validation, can be found in [26].

In this section the simulation results of the FCU model as simulated with both methods are presented and compared. The FCU performance variables as well as the solver function variables are shown in order to compare the performance of the two methods. Room air temperature and cooling coil valve position are the FCU performance variables. The number of function evaluations, number of iterations, cumulative number of iterations, and cumulative number of function evaluations are the performance variables of each solver method.

In Figs. 4 and 5 the FCU performance variables of simulations for east and south facing rooms using both the PH and LM methods are compared for identical initial conditions. The solid line represents experimental data, the dotted line represents the simulation result with the PH method and dashed line represents simulation results with the LM method. The FCU model parameters have been validated using experimental data which are reported in another publication [26]. Here, experimental data serve as a reference to compare the two solver methods.

As Fig. 4 illustrates, the FCU in the east room is operated in cooling mode to maintain the cooling temperature set point. The cooling coil valve controls the water flow rate, which in part determines the heat transfer rate, and maintains the room temperature near the cooling set point. As the east room air temperature graph in Fig. 4 shows, the results of the simulation for the PH and LM methods for room air temperature are identical except for the 10:00 to 11:06 and 13:30 to 15:46 time frames, in which the model using the PH method could not produce a room temperature that stabilizes at the cooling temperature set point. Further analysis of these two time frames shows unsuccessful convergences (unsuccessful iteration times have been listed in Table 4). When a nonlinear system does not converge at a given time step (i.e., the simulation fails to meet the termination criteria), the inaccurate solution is passed to later time steps. This can lead the simulation into non-physical state space (e.g., negative humidity) and/or an inaccurate solution. This can happen when residuals become small, but fail to vanish. Thus, while all but one superblock may reach convergence within a time step, the one superblock that failed to converge can cause the entire simulation to fail at that time step. That scenario is avoided by the LM method, which successfully iterates all superblocks to desirable solutions within the simulation time window. For this reason, the simulation result for cooling coil valve position by the LM method is reasonable, whereas the solution provided by the PH method is not. The bump that occurs around 11:00 in the east room air temperature values calculated by the PH method shows that solver became trapped in an unsatisfactory region where it could not find the minimum of the sum of the squares of the functions.

As Fig. 5 shows, the FCU operation in the south room is consistent with the operation in the east room. The PH method did not solve the thermal superblock in the 10:00–11:06 and 13:12–15:43 time frames, having completed most of iterations unsuccessfully which have been listed in Table 5. On the other hand, the LM method converged to a desirable solution for both the air flow and thermal

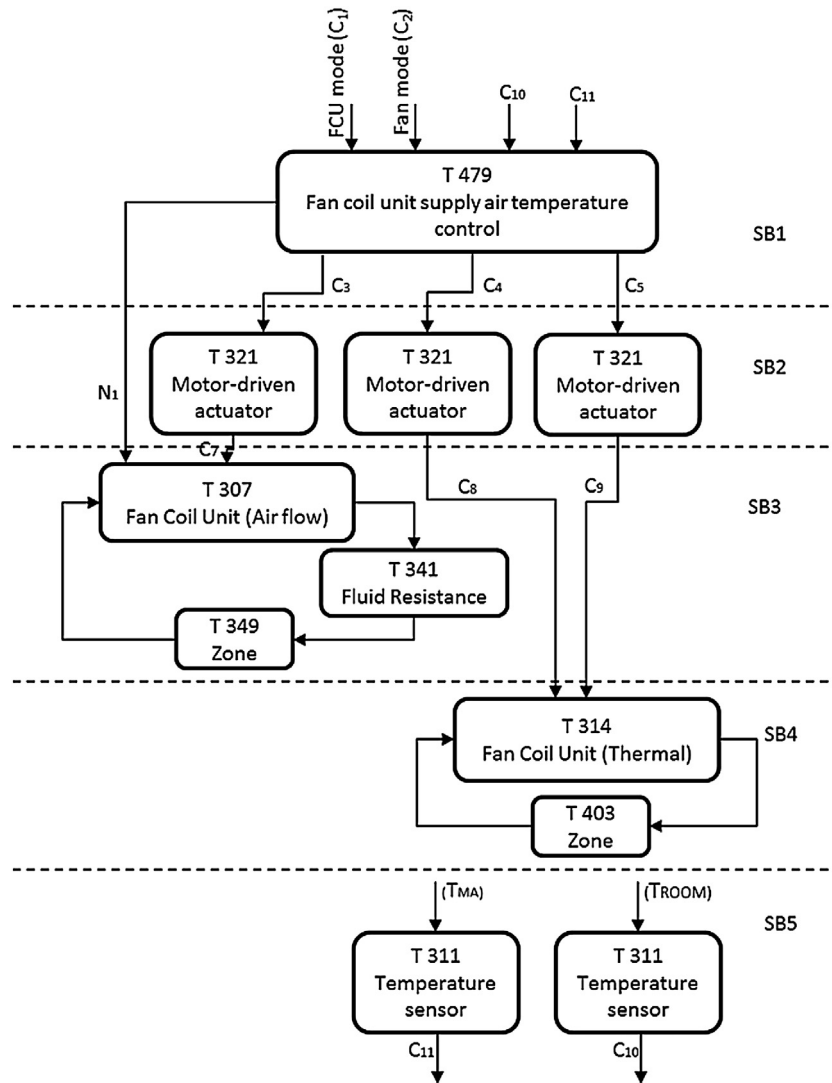


Fig. 3. Fan coil unit model structure in HVACSIM+.

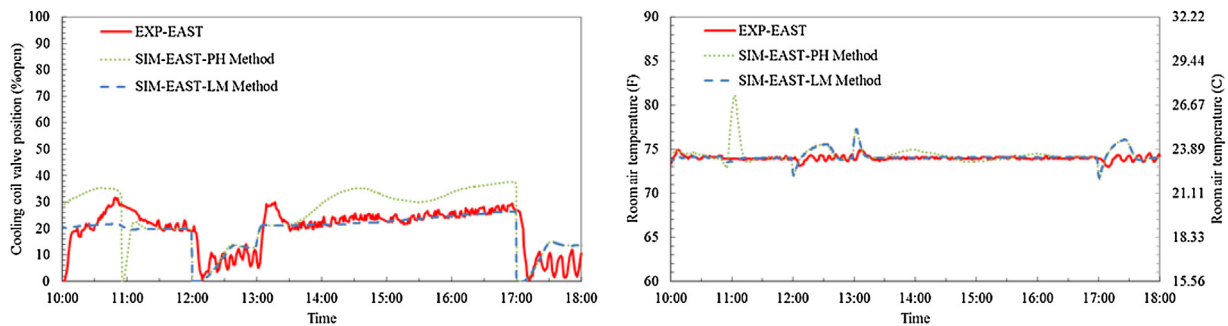


Fig. 4. Simulation results of FCU operational variables in east room by PH and LM method.

superblocks. The simulation results predicting the FCU operation in the east and south rooms are acceptable when the solver returns a converged solution for all superblocks. As the south room air temperature graph in Fig. 5 shows, the results of the simulation for the PH and LM methods for room air temperature are identical except for the 10:00–11:10, 13:10–14:00 and 14:30–16:00 time frames, in which the PH method moves around the cooling temperature set point but is not able to capture it. The cooling coil valve

position based on the LM method, which successfully converged to a solution, gives more reasonable results.

The prior validation work [26] revealed some weaknesses in the combined model of the FCU and room. The thermodynamic interaction between the modeled room and the ambient environment differs from the interaction that occurs in the real physical space. Thus, the cooling coil valve signals solved for in the control superblock do not completely agree with the experimental data. This weakness in the model is most clearly demonstrated by the

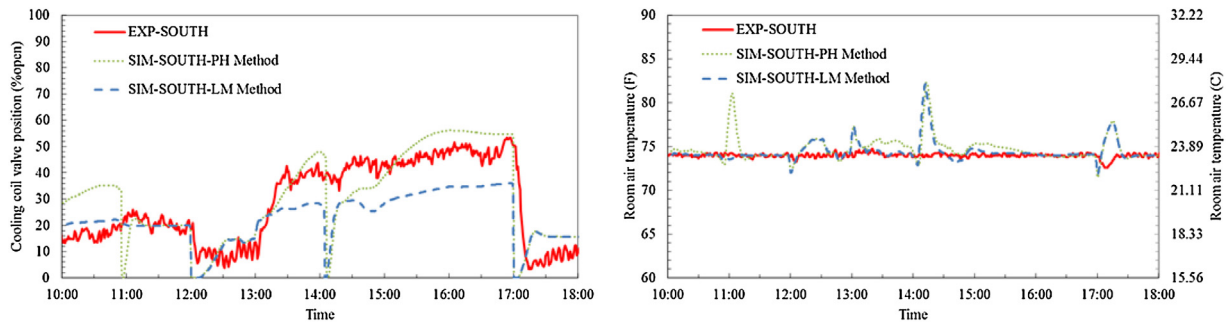


Fig. 5. Simulation results of FCU operational variables in south room by PH and LM method.

Table 4

Unsuccessful iteration times for east room model using PH method.

| Time (minute) | Time (h:mm) | Time (minute) | Time (h:mm) |
|---------------|-------------|---------------|-------------|
| 603.1334 | 10:03 AM | 846.1667 | 2:06 PM |
| 606.1166 | 10:06 AM | 846.4167 | 2:06 PM |
| 607.0667 | 10:07 AM | 850.6667 | 2:11 PM |
| 619.7833 | 10:20 AM | 851.2833 | 2:11 PM |
| 625.3 | 10:25 AM | 852.2167 | 2:12 PM |
| 628.0833 | 10:28 AM | 853.75 | 2:14 PM |
| 630.7833 | 10:31 AM | 858.55 | 2:18 PM |
| 632.55 | 10:32 AM | 860.4833 | 2:20 PM |
| 646.3834 | 10:46 AM | 865.5 | 2:25 PM |
| 650.25 | 10:50 AM | 871.0667 | 2:31 PM |
| 650.5833 | 10:50 AM | 871.1334 | 2:31 PM |
| 651.05 | 10:51 AM | 873.65 | 2:34 PM |
| 653.32 | 10:53 AM | 876.0833 | 2:36 PM |
| 654.09 | 10:54 AM | 877.3 | 2:37 PM |
| 656.96 | 10:57 AM | 878.8333 | 2:39 PM |
| 658.86 | 10:59 AM | 881.85 | 2:42 PM |
| 660.03 | 11:00 AM | 889.1667 | 2:49 PM |
| 661.32 | 11:01 AM | 889.65 | 2:50 PM |
| 663.25 | 11:03 AM | 914.0833 | 3:14 PM |
| 665.65 | 11:06 AM | 917.7 | 3:18 PM |
| 809.3 | 1:29 PM | 920.3167 | 3:20 PM |
| 812.3834 | 1:32 PM | 920.7 | 3:21 PM |
| 818.4667 | 1:38 PM | 921.1334 | 3:21 PM |
| 823.65 | 1:44 PM | 923.6166 | 3:24 PM |
| 825.6334 | 1:46 PM | 923.6833 | 3:24 PM |
| 827.45 | 1:47 PM | 931.9833 | 3:32 PM |
| 836.9333 | 1:57 PM | 940.2833 | 3:40 PM |
| 841.2667 | 2:01 PM | 946.2333 | 3:46 PM |
| 841.35 | 2:01 PM | | |

Table 5

Unsuccessful iteration times for south room modeled by PH method.

| Time (minute) | Time (h:mm) | Time (minute) | Time (h:mm) |
|---------------|-------------|---------------|-------------|
| 600.4333 | 10:00 AM | 801.4667 | 1:21 PM |
| 601.95 | 10:01 AM | 812.2667 | 1:32 PM |
| 606.7667 | 10:06 AM | 816.0333 | 1:36 PM |
| 610.2333 | 10:10 AM | 817.9667 | 1:38 PM |
| 610.7167 | 10:10 AM | 821.25 | 1:41 PM |
| 612.65 | 10:12 AM | 825.4 | 1:45 PM |
| 615.4833 | 10:15 AM | 827.6 | 1:48 PM |
| 616.2333 | 10:16 AM | 832.5833 | 1:52 PM |
| 621.45 | 10:21 AM | 861.6833 | 2:22 PM |
| 625.95 | 10:25 AM | 883.85 | 2:44 PM |
| 628.0833 | 10:28 AM | 889.55 | 2:49 PM |
| 634.3167 | 10:34 AM | 900 | 3:00 PM |
| 635.8834 | 10:36 AM | 900.1833 | 3:00 PM |
| 638.1334 | 10:38 AM | 906.55 | 3:06 PM |
| 638.7 | 10:39 AM | 908.0667 | 3:08 PM |
| 642.6334 | 10:43 AM | 911.7833 | 3:12 PM |
| 646.6833 | 10:47 AM | 911.8333 | 3:12 PM |
| 652.6 | 10:53 AM | 913.5833 | 3:13 PM |
| 656.96 | 10:54 AM | 915.7333 | 3:16 PM |
| 658.86 | 10:57 AM | 916.4833 | 3:16 PM |
| 660.03 | 10:59 AM | 922.7333 | 3:23 PM |
| 661.32 | 11:00 AM | 924.75 | 3:25 PM |
| 663.25 | 11:01 AM | 927.9 | 3:28 PM |
| 665.65 | 11:03 AM | 928.15 | 3:28 PM |
| 790.7833 | 11:06 AM | 934.8666 | 3:35 PM |
| 792 | 1:12 PM | 942.2 | 3:42 PM |
| 794.6334 | 1:15 PM | 942.65 | 3:43 PM |
| 799.25 | 1:19 PM | | |

decrease in cooling coil valve position shown near 14:00 in Fig. 5 for both the PH and LM methods.

Figs. 6–9 are a brief comparison of the PH and LM method performance variables obtained in solving the system of nonlinear equations that emerge in the FCU simulation. Fig. 6 is a comparison of the PH and LM method cumulative number of iterations and the number of iterations at each time step in the air flow and thermal superblocks for the FCU in the east room. Similarly, Fig. 7 displays the comparison of the PH and LM method cumulative number of function evaluations and the number of function evaluations at each time step in the air flow and thermal superblocks for the FCU in the east room; Figs. 8 and 9 display the same data for the south room. According to the number of iterations at each time step for the thermal superblock of the FCU in the east and south room in Figs. 6 and 8, it is obvious that for the PH method, during the time frames with unsuccessful solutions, the number of iterations in the thermal superblock is greater. This implies that the PH method attempts to make good progress toward the solution by increasing the number of iterations. A close look at the number of function evaluations in Figs. 7 and 9 for the FCU thermal superblock in the east and south rooms leads us to the same conclusion. Note that in Fig. 9, near 14:00 the number of function evaluations does

not spike for either PH or LM because, as noted above, the deviation from the experimental data is not due to convergence failure.

Figs. 7 and 9 show that the LM method requires more function evaluations than the PH method because, as discussed in Section 2, the LM method solves the normal equations directly rather than using the approximation employed in the PH method. Figs. 6 and 8 show that, for the airflow superblock, the LM method requires more iterations than the PH method; this is also a result of the direct solution of the normal equations. In Fig. 8, for the thermal superblock, the LM method requires more iterations than the PH method, as expected, but in Fig. 6, the thermal superblock requires more iteration for the PH method than for the LM method. This behavior demonstrates the difficulty that the PH method has converging to a solution in the thermal superblock for the east room as previously discussed in relation to Fig. 4. If both methods are properly converging, the LM method should require more iterations than PH due to its greater complexity.

4. Conclusion

A qualitative difference between the LM and PH methods arises when a trial step fails to yield sufficient decrease in the residual of the nonlinear equations and the search direction must be

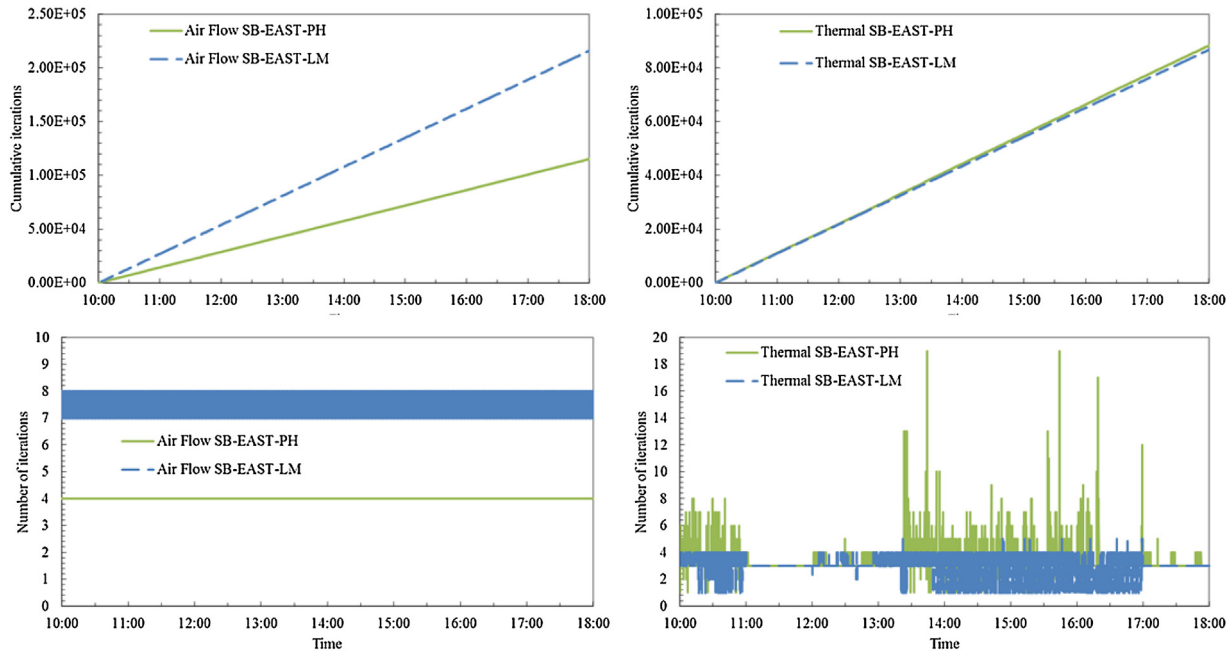


Fig. 6. Comparison of the cumulative number of iterations and the number of iterations at each time step for the PH and LM methods in the air flow and thermal superblocks for the FCU in the east room. In the air flow superblock the LM method rapidly oscillates between seven and eight iterations, resulting in what appears to be a solid block.

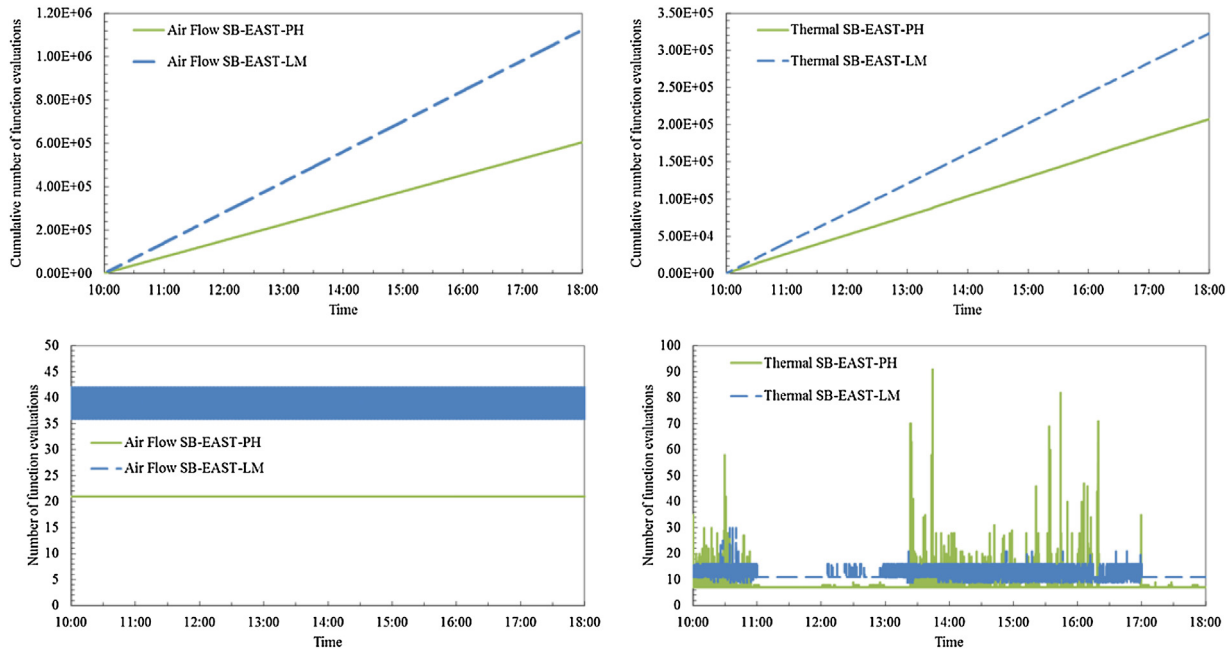


Fig. 7. Comparison of the cumulative number of function evaluations and the number of function evaluations at each time step for the PH and LM methods in the air flow and thermal superblocks for the FCU in the east room. In the air flow superblock the LM method rapidly oscillates between seven and eight iterations, resulting in what appears to be a solid block.

altered. Many simple methods for solving nonlinear equations simply shorten the length of the search trial step in hopes that a shorter step will yield sufficient decrease of the residuals. Other more sophisticated techniques alter both the length and the direction of the trial step. In the case of LM, the μ parameter is increased and the resulting Equations 6 is solved. This is the case even when a trial step with length less than that of the Cauchy step yields sufficient decrease in the residuals of the nonlinear equations and this trial step is an acceptable step. On the other hand, when PH computes a trial step, it is generated by augmenting or ‘adding onto’ the Gauss-Newton step. Once the Gauss-Newton step has been computed at

a given point, the PH algorithm computes trial steps by changing the parameter Π . The PH algorithm, therefore, computes all trial steps, both successful steps and failed steps, without resolving the normal equations, (Eq. (7)). This is true even when a trial step with length less than the length of the Cauchy step is chosen to be the accepted step of PH. Where the LM method solves the normal equations, the PH method uses an approximation. This is the reason for the improvements in the LM method over the PH method, but it is also the reason why the LM method can be more computationally expensive.

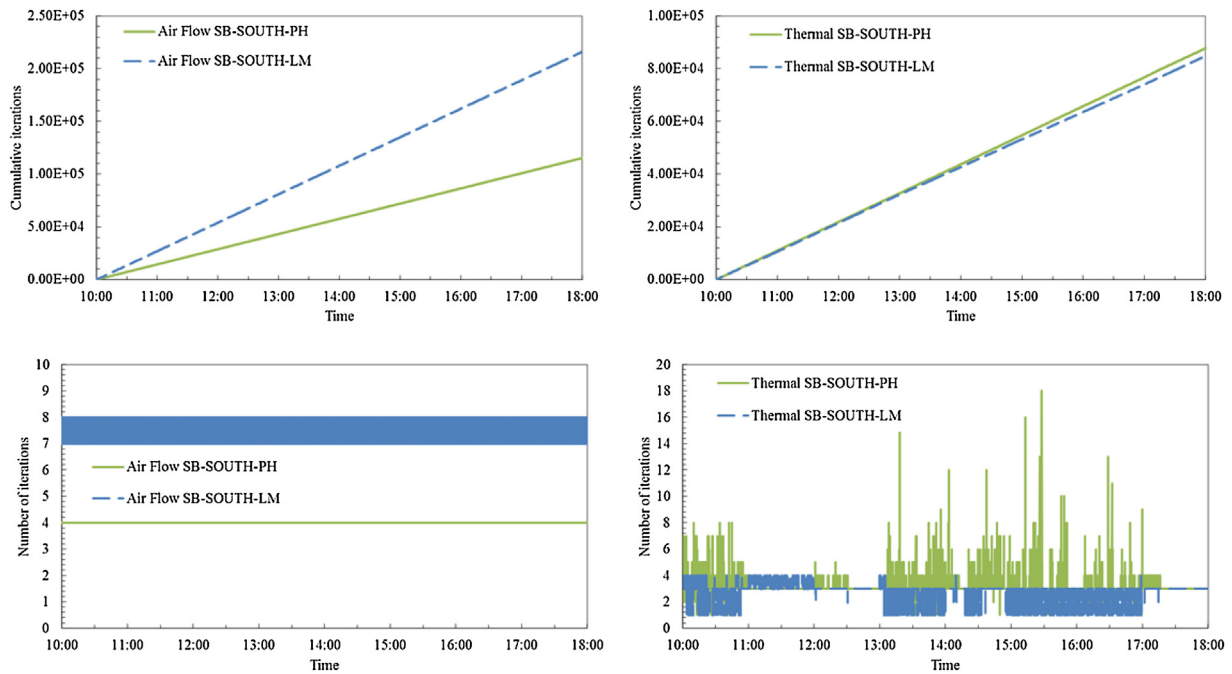


Fig. 8. Comparison of the cumulative number of iterations and the number of iterations at each time step for the PH and LM methods in the air flow and thermal superblocks for the FCU in the south room. In the air flow superblock the LM method rapidly oscillates between seven and eight iterations, resulting in what appears to be a solid block.

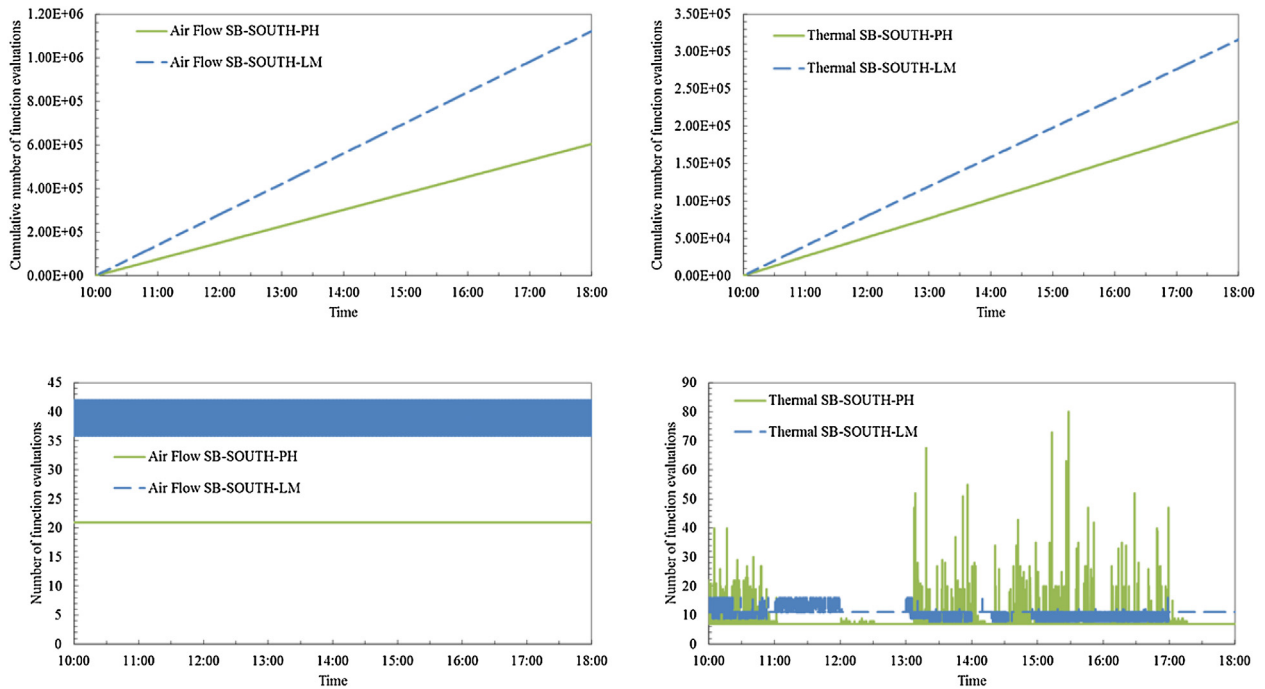


Fig. 9. Comparison of the cumulative number of function evaluations and the number of function evaluations at each time step for the PH and LM methods in the air flow and thermal superblocks for the FCU in the south room. In the air flow superblock the LM method rapidly oscillates between seven and eight iterations, resulting in what appears to be a solid block.

An important computational issue that arises in the multi-physics framework of modeling buildings is that of preconditioning of nonlinear systems. This occurs when two or more component models are coupled with the goal of simulating events involving the output from these components. Many linear preconditioners exist for applications-specific linear systems; however, in the context of improving HVACSIM+, greater improvement can be seen from

considering directly the nonlinear system coupling components. Future research will focus on improving the problem formulation by developing and applying a preconditioner that is designed specifically for HVAC applications. This process will include creating a more uniform scaling across variables and ordering the solution of equations in a way that is numerically stable.

Acknowledgement

The authors would like to thank National Institute of Standards and Technology for funding the project under grant number: 60NANB10D243.

References

- [1] DOE, Building Energy Software Tools Directory, U.S. Department of energy, Available at: <http://www.eere.energy.gov/> (accessed August, 2009) (2009).
- [2] D.B. Crawley, J.W. Hand, M. Kummert, B.T. Griffith, Contrasting the capabilities of building energy performance simulation programs, *Build. Environ.* 43 (2008) 661–673.
- [3] J. Hensen, Simulation Tools for Energy Efficient Building Design, Available at: <http://www.bwk.tue.nl/bps/hensen/courseware> (accessed August, 2009) (2009).
- [4] M. Trcka, J.L.M. Hensen, Overview of HVAC system simulation, *Autom. Constr.* vol. 19 (2) (2010) 93–99.
- [5] SPARK, Simulation Problem Analysis and Research Kernel, Lawrence Berkeley National Laboratory and Ayres Sowell Associates, Inc., 2003.
- [6] EnergyPlus, EnergyPlus Energy Simulation Software, U. S. Department of Energy, 2005 <http://www.eere.energy.gov/buildings/energyplus/>.
- [7] EnergyPlus, The Reference to EnergyPlus Calculations, U. S. Department of Energy, 2014.
- [8] LBNL, Modelica Library for Building Energy and Control Systems, Lawrence Berkeley National Laboratory, 2011 <https://gaia.lbl.gov/bir>.
- [9] P. Frittzson, Principals of Object-Oriented Modeling and Simulation with MODELICA 2.1, IEEE Press, 2004, pp. 667–677.
- [10] SEL, TRNSYS Version 16 Programmer's Guide, vol. 8, Solar Energy Laboratory, Mechanical Engineering Department, University of Wisconsin, Madison, 2007 (February).
- [11] B. Carnahan, H.A. Luther, J.O. Wilkes, *Applied Numerical Methods*, John Wiley & Sons, New York, 1969.
- [12] C. Park, D.R. Clark, G.E. Kelly, An overview of HVACSIM+, a dynamic building/HVAC/Control systems simulation program, in: *Building Energy Simulation Conference*, Seattle, Washington, August, 1985, pp. 21–22.
- [13] D.R. Clark, W.R. May Jr., HVACSIM+ building systems and equipment simulation program users guide, NBSIR 85–3243, Natl. Bur. Stand. (1985).
- [14] C. Park, D.R. Clark, G.E. Kelly, HVACSIM+ building systems and equipment simulation program building loads calculation, NBSIR 86–3331, NIST (1986).
- [15] D.R. Clark, HVACSIM+ building systems and equipment simulation program reference manual, NBSIR 84–2996, NIST (1985).
- [16] P. Haves, L.K. Norford, A standard simulation testbed for the evaluation of control algorithms and strategies, in: ASHRAE RP-825 Final Report, American Society of Heating, Refrigerating and Air Conditioning Engineers Inc., Atlanta, 1997.
- [17] J. Wen, S. Li, Tools for evaluating fault detection and diagnostic methods for air-handling units, in: ASHRAE RP-1312 Final Report, American Society of Heating, Refrigerating and Air Conditioning Engineers Inc., Atlanta, 2011.
- [18] A. Shterenlikht, N.A. Alexander, Levenberg-Marquardt vs Powell's dogleg method for Gurson-Tvergaard-Neeldeman plasticity model, *Comput. Methods Appl. Mech. Eng.* 23 (2012) 7–240.
- [19] K. Levenberg, A method for the solution of certain non-linear problems in least squares, *Quarterly Appl. Math.* 2 (2) (1944) 164–168.
- [20] D. Marquardt, An algorithm for the least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* 11 (2) (1963) 431–441.
- [21] M.J.D. Powell, A hybrid method for nonlinear equations, in: P. Rabinowitz (Ed.), *Numerical Methods for Nonlinear Algebraic Equations*, Gordon and Breach Science, London, 1970, pp. 87–144.
- [22] M.J.D. Powell, A new algorithm for unconstrained optimization, in: J.B. Rosen, Ol. L. Mangasarian, K. Ritter (Eds.), *Nonlinear Programming*, Academic Press, New York, 2016, p. 1970.
- [23] N. Yamashita, M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, *Computing (Suppl. 15)* (2001) 237–249.
- [24] C.T. Kelley, *Iterative Methods for Optimization*, Frontiers in Applied Mathematics, vol. 18, SIAM, Philadelphia, PA, 1999.
- [25] G.W. Stewart, J.G. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, 1990.
- [26] S. Pourarian, J. Win, D. Veronica, X. Zhou, R. Lui, A Tool for Evaluating Fault Detection and Diagnostic Methods for Fan Coil Units. *Annual ASHRAE Conference*, Seattle, Washington, June 28–July 2, 2014.