Scaling Methods for Dynamic Building System Simulation in an HVACSIM+ environment

Zhelun Chen[1], Jin Wen[1], Anthony Kearsley[2], Amanda Pertzborn[2]
[1]Drexel University, Philadelphia, PA, United States
[2]National Institute of Standards and Technology, Gaithersburg, MD, United States

Abstract

A robust and efficient nonlinear algebraic and differential equations solver is increasingly essential as the scale of dynamic building system simulation grows larger and the models employed become more complex. There are many factors affecting the performance of these solvers. In this paper the impact of variable scaling on the convergence of a family of Newton-method based solvers is summarized. A comparison of the performance of a few different scaling methods is presented using HVACSIM+, a software simulation tool that employs Powell's Hybrid method to solve the system of nonlinear algebraic equations. A small commercial building is modelled in HVACSIM+ and serves as an illustrative example for this study. It is demonstrated that, for these methods, variable scaling can strongly influence computational performance but, for our example problem, any improvement or deterioration in numerical accuracy is limited. The conclusions are not limited to the software platform HVACSIM+.

Introduction

Background

The building sector represents the largest primary energy-consuming sector in the United States, responsible for 41% of the country's primary energy, in comparison to 28% for the transportation sector and 32% for industry (DOE, 2011). Moreover, buildings consume 74% of the electricity in the United States, which makes the building sector significant to the overall smart grid infrastructure. Given the rapid development of the smart grid and the potential, through demand shifting, of buildings to store and generate electricity (DOE, 2011), there is an urgent need to improve the dynamic interactions between buildings and the smart grid. Accomplishing this task requires an ability to robustly simulate building dynamics on time scales of seconds. Unlike traditional dynamic building simulations, smart-city and smart grid dynamic building simulations will need to simulate large and complex building systems and building clusters that include multiple buildings that are coupled to various degrees, either through the smart grid or other means, such as district heating/cooling. This presents a unique and very difficult computational challenge. For example, large systems of coupled, potentially ill-conditioned, nonlinear equations, easily reaching thousands, will need to be solved rapidly and robustly.

Currently, four commonly-used and publicly available simulation tools exist to simulate the dynamic behaviour of fully integrated building and HVAC systems, on a time scale of seconds: HVACSIM+, TRNSYS, SPARK, and MODELICA. Table 1 summarizes the nonlinear algebraic solvers employed by each of these four simulation tools (Park et. al. , 1986, Klein et. al. , 2004, SPARK, 2003, Fritzson, 2014).

Table 1 Building Dynamic Simulation Software and Their Solvers

Software	Solver
HVACSIM+	Powell's Hybrid Method
TRNSYS	Successive Substitution Method; Powell's Hybrid Method

SPARK	Newton Raphson's Method;
	Perturbed Newton Method;
	Successive Substitution Method;
	Secant Method

MODELICA	Newton Raphson's Method

Among the myriad of factors that can affect a nonlinear solver's performance (as measured by accuracy and computational cost), the current study will examine the impact of matrix scaling methods on the convergence of the solver using HVACSIM+ as the platform (though the conclusions can be extended to any Jacobian based solver). This study constitutes part of an ongoing study of the numerical performance of nonlinear solvers and their variations when applied to this class of problems. Uniform scaling across variables is known to be crucial to the performance of general nonlinear solvers (Dennis and Schnabel, 1996, Pourarian et. al. , 2016).

HVACSIM+, employs a dense matrix implementation of Powell's Hybrid Method (Powell, 1970), a very commonly employed method for problems typically encountered when performing simulation in the building domain. This paper introduces HVACSIM+ in the first section. A description of the scaling method study and the simulation testbed follow in the next section. The paper concludes with test results and observations.

HVACSIM+

Developed by the U.S. National Institute of Standards and Technology (NIST), HVACSIM+ is a dynamic component-based simulation tool and computational environment (Park et. al., 1986) that includes a collection of subroutines in three categories: pre-processing,

https://doi.org/10.26868/25222708.2017.534

simulation, and post-processing. During the pre-processing stage, a simulation work file is created by use of the interactive front-end program. The simulation work file is then converted to a model definition file to make the model information readable by the main simulation program, MODSIM.

MODSIM employs a hierarchical structure in the simulation, illustrated in Figure 1. In this structure, a UNIT is an individual instance of a generic component model representing a specific component in the system. The equations that simulate a physical component (such as a fan) are contained in a subroutine called TYPE. During the model construction, a UNIT calls a TYPE and provides detailed parameters of a specific component to the TYPE to simulate the component. Users are required to link each UNIT by its inputs and outputs to represent the relationships in the real physical system. Closely-coupled UNITs are grouped by the user into blocks for simultaneous solution. Blocks are then similarly grouped into a superblock for simultaneous solution. State variables in different superblocks are weakly coupled and each superblock is treated as an independent subsystem of the overall simulation (Park et. al., 1986).

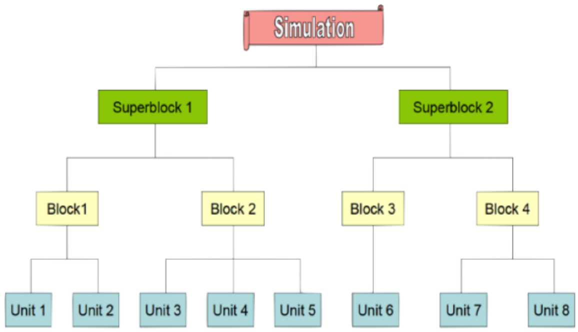

Figure 1 Hierarchical simulation setup

Equations in a block or superblock are solved numerically by the subroutine SNSQ (Hiebert, 1982) and its associated subroutines, which are an implementation of a nonlinear equation-solving method based on the Powell's Hybrid method (Powell, 1970). Typical Newton methods can reach local convergence rapidly, however, they require a good initial guess to reach that solution. In order to improve the convergence without losing the efficiency of Newton's method, Powell suggests a trust-region method that combines Gauss-Newton and steepest descent methods.

The purpose of SNSQ is to solve a set of (n) nonlinear equations, i.e.,

$$\boldsymbol{F}(\boldsymbol{x}) = 0, \qquad \boldsymbol{F}: \mathbb{R}^n \to \mathbb{R}^n \tag{1}$$

where $\boldsymbol{F}$ is assumed to be a continuously differentiable function everywhere in $\mathbb{R}^n$.

In the trust-region method employed by Powell, the objective function, $\boldsymbol{F}$ is used to construct a quadratic model function, $\boldsymbol{H}$, so that in a neighborhood of the current iteration at which $\boldsymbol{H}$ is constructed, the functions $\boldsymbol{H}$ and $\boldsymbol{F}$ are similar. The function $\boldsymbol{H}$ is said to be trusted to accurately model $\boldsymbol{F}$ in a region with weighted radius Π that is centred at the current iterate. The constructed quadratic model function is:

$$\boldsymbol{H}(\boldsymbol{s}^{(k)}) = \left\|\boldsymbol{F} + \boldsymbol{J}\boldsymbol{s}^{(k)}\right\|^2 \tag{2}$$

where $\boldsymbol{F}$ and $\boldsymbol{J}$ are the objective function value and Jacobian at the current iterate $\boldsymbol{x}^{(k)}$, and $\boldsymbol{s}^{(k)}$ is the step moving from the current iterate $\boldsymbol{x}^{(k)}$ to the next iterate $\boldsymbol{x}^{(k+1)}$. The Euclidean norm, denoted by, $\|*\|$, is used in Eq. (2). In this paper, only the Euclidean norm is used and it will be denoted as norm.

In practice, the trust-region radius is selected based on the ability of the model to "fit" the objective function. This ability is measured by the ratio of the reduction of the objective function (actual reduction) to the reduction of the constructed quadratic function (predicted reduction), as shown in Eq. (3).

$$r^{(k)} = \frac{\left\|\boldsymbol{F}(\boldsymbol{x}^{(k)})\right\|^2 - \left\|\boldsymbol{F}(\boldsymbol{x}^{(k+1)})\right\|^2}{\boldsymbol{H}(\boldsymbol{0}) - \boldsymbol{H}(\boldsymbol{s}^{(k)})}. \tag{3}$$

A strong agreement ($r^{(k)}$ closer to 1) between the objective and constructed function suggests a strong value of the trust-region radius Π, and leads to an expansion of the trust-region at the next iteration. Conversely, a weak agreement leads to a shrinkage of the trust-region.

Powell's Hybrid method first seeks to attain the Gauss-Newton step, $\boldsymbol{s}_{gn}{}^{(k)}$, by solving the normal equation. If $\left\|\boldsymbol{s}_{gn}{}^{(k)}\right\|$ is within Π, it will be accepted as the step for this iteration. If $\boldsymbol{s}_{gn}{}^{(k)}$ is not accepted, a gradient step, $\boldsymbol{s}_{sd}{}^{(k)}$, which is unique in the steepest descent (or gradient) direction, is calculated by solving,

$$\min_{\boldsymbol{s}^{(k)}} \boldsymbol{H}(\boldsymbol{s}^{(k)}) \quad \text{subject to } \left\|\boldsymbol{s}^{(k)}\right\| \le \Pi \tag{4}$$

An illustration of the step finding process by Powell's Hybrid method can be seen in Figure 2.

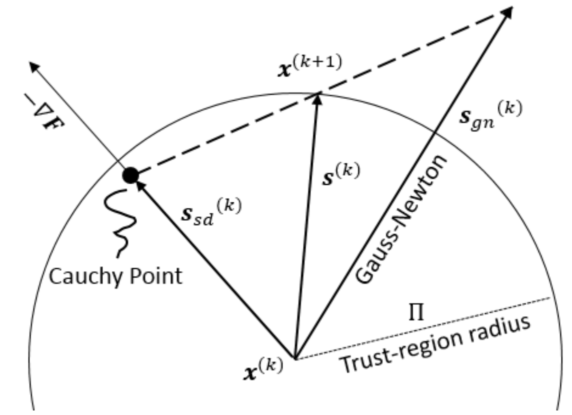

Figure 2 Visualization of Step Finding Process by Powell's Hybrid's Method [Reproduction of Fig.1 in (Pourarian et. al., 2016)]

The point at $\boldsymbol{x}^{(k)} + \boldsymbol{s}_{sd}{}^{(k)}$ is called the "Cauchy Point". The (hybrid) step, $\boldsymbol{s}^{(k)}$, for the current iteration is then determined as the linear combination between the Gauss-

Newton step, $\boldsymbol{s}_{gn}{}^{(k)}$, and the gradient step, $\boldsymbol{s}_{sd}{}^{(k)}$, such that the norm of the step is equal to the trust-region radius.

Scaling Methods

In building system simulation, there may be substantial variation in the magnitude of variables both when starting a simulation and upon completion of a simulation. For example, the magnitude of temperature is usually in the range of $[10^0, 10^2]$ °C, while the humidity ratio is in the range of $[10^{-3}, 10^{-2}]$ kg/kg. These variables can change drastically in the course of a single simulation. If variables have similar magnitude, then the resulting system that must be solved to generate a search direction is typically well-conditioned. When a collection of variables resides on hugely varying scales, many orders of magnitude difference between them, then the resulting linear system that must be solved can be very poorly scaled leading to longer solution times, less accurate solutions to linear systems and excessive damping of a search direction to achieve global convergence.

Consider the diagonal scaling matrix $\boldsymbol{D}_{\boldsymbol{x}}$ that transforms the variable space from $\boldsymbol{x}$ to $\widetilde{\boldsymbol{x}}$, and the function,

$$\widetilde{\boldsymbol{F}}(\widetilde{\boldsymbol{x}}) = \boldsymbol{F}\left(\boldsymbol{D}_{\boldsymbol{x}}{}^{-1}\,\widetilde{\boldsymbol{x}}\right) = 0, \qquad \widetilde{\boldsymbol{F}}: \mathbb{R}^n \to \mathbb{R}^n \tag{5}$$

is to be solved instead of Eq. (1). By this transformation, the Gauss-Newton step is unaffected but the steepest descent direction is changed. Benefits of employing variable scaling, if variables are properly scaled, can be seen not only from a better conditioned steepest descent direction, but also from the reduction of rounding errors due to addition or subtraction in a finite precision computer. $\boldsymbol{D}_{\boldsymbol{x}}$ can be taken to be the appropriately sized identity matrix if no scaling is employed,

$$\boldsymbol{D}_{\boldsymbol{x}} = \boldsymbol{I}. \tag{6}$$

In HVACSIM+, initial scaling takes place at the beginning of every simulation time step and the procedure seeks to improve the conditioning of the linear system resulting from Eq. (5). Three commonly employed scaling methods were incorporated into HVACSIM+ for this comparative study:

<u>Method 1:</u>

$$\boldsymbol{D}_{\boldsymbol{x}}(j) = \frac{1}{\sqrt{[\boldsymbol{x}_j{}^{(0)}]^2 + tol^2}} \tag{7}$$

where $\boldsymbol{x}_j{}^{(0)}$ is the initial value of the jth state variable at each time step and tol is a selected tolerance.

<u>Method 2:</u>

$$\begin{cases} \boldsymbol{D}_{\boldsymbol{x}}(j) = \left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\|, & \text{if } \left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\| \neq 0 \\ \boldsymbol{D}_{\boldsymbol{x}}(j) = 1, & \text{if } \left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\| = 0 \end{cases} \tag{8}$$

<u>Method 3:</u>

$$\begin{cases} \boldsymbol{D}_{\boldsymbol{x}}(j) = \dfrac{1}{\left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\|}, & \text{if } \left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\| \neq 0 \\ \boldsymbol{D}_{\boldsymbol{x}}(j) = 1, & \text{if } \left\|\boldsymbol{J}(\boldsymbol{x})_{col(j)}\right\| = 0 \end{cases} \tag{9}$$

In method 2 and method 3, $\boldsymbol{J}(\boldsymbol{x})_{col(j)}$ is the jth column of the Jacobian matrix evaluated at the initial guess of parameter values at each time step.

In HVACSIM+, a standard forward centered finite difference numerical approximation to the Jacobian is employed as analytical expressions for derivatives are rarely available. It is worth noting that in our numerical experimentation we have seen strong evidence of structured variation in control variables. That is to say, variables that yield more sensitivity to output can be separated during a finitie difference procedure and treated differently than those variables having less or little effect on output.

There exist many suggestions to select finite difference step length to obtain better approximations to the Jacobian, a step length that seeks to perturb the maximum number of digits in a given finite precision arithmetic is often used (Dennis and Schnabel, 1996),

$$h_\varepsilon = \sqrt{\varepsilon_{mach}} \cdot sign(x_j) \tag{10}$$

where ε_{mach} is the machine precision.

Pernice and Walker (Pernice and Walker, 1998) suggested

$$h_{PW} = \sqrt{(1 + \|\boldsymbol{x}\|)\varepsilon_{mach}} \cdot sign(x_j), \tag{11}$$

and Dennis and Schnabel (Dennis and Schnabel, 1996) suggested a finite difference step length that depends on the magnitude of each state variable,

$$h_{DS} = \begin{cases} |x_j| \cdot h_\varepsilon, & \text{if } x_j \neq 0 \\ h_\varepsilon, & \text{if } x_j = 0 \end{cases} \tag{12}$$

The finite difference step length described in Eq. (12) is currently employed by HVACSIM+.

In this study, the three scalings, method 1, method 2, and method 3, are examined, and their dependence on the Jacobian is studied by varying the finite difference step length in the approximation of the Jacobian.

Numerical Experiments

Simulation Testbed

The numerical testbed for this study is a small commercial building simulation model developed as part of ASHRAE research project 1312 (Wen and Li, 2011). This model simulates a single duct dual fan variable air volume system serving four building zones. There are ten superblocks in this model, among which three superblocks contain variables that are solved simultaneously within the specific superblock. These superblocks are the control superblock, airflow superblock, and thermal superblock. Table 2 summarizes the number of simultaneous variables and the category of those variables in each superblock.

Test Cases

Eight test cases with different scaling methods and finite difference step length calculation methods are summarized in Table 3. The eight test cases are implemented in the numerical testbed for three seasons:

spring, summer and winter, among which the system operating conditions are different.

Table 2 Category and number of simultaneous variables in each superblock

Superblock	Control	Airflow	Thermal
Category	Control Signal	Mass Flow Rate; Pressure	Temperature; Humidity
Number	7	37	49

Table 3 Test Cases (Eq. No. in parenthesis)

Case	**Scaling Method**	**Finite Difference Step Length**
1	No Scaling: (6)	Default Method: (12)
2	Method 1: (7)	Default Method: (12)
3	Method 2: (8)	Default Method: (12)
4		Simple Method: (10)
5		Pernice & Walker: (11)
6	Method 3: (9)	Default Method: (12)
7		Simple Method: (10)
8		Pernice & Walker: (11)

Simulation Settings

For each season, the duration of each test is 24 hours and the simulation time step is fixed as 2.5 seconds. During the 24-hour period, the HVAC system operates from 6 AM to 6 PM. The same convergence criteria are employed in all of the numerical tests. The solution to which the numerical method converged is considered correct if it passes either one of the following criteria:

Convergence Criterion 1:

Convergence based on the size of the trust region, which indicates change in variables between two iterations, is less than or equal to the tolerance, as shown in Eq. (13),

$$\Pi \leq xtol \cdot \left\| \boldsymbol{D}_x \cdot \boldsymbol{x}^{(k+1)} \right\|. \tag{13}$$

Convergence Criterion 2:

Convergence based on the norm of the residual being less than or equal to a prescribed tolerance, as shown in Eq. (14),

$$\left\| \boldsymbol{F}(\boldsymbol{x}^{(k+1)}) \right\| \leq ftol. \tag{14}$$

Tolerances are taken to be $xtol = 3\times10^{-5}$ and $ftol = 5\times10^{-5}$ for all numerical results presented here.

A third condition, which indicates a lack of convergence, is triggered if the number of function evaluations reaches a pre-determined limit. In the numerical results presented here this was $200 \cdot (n+1)$, where n is the number of equations to be solved.

Results and Discussions

In this section, only results associated with summer conditions are presented because the system operating conditions in summer demonstrate more dynamics (i.e., more frequent and larger damper and valve movements) than in the other seasons, making the impact of different scaling methods more observable. The termination conditions, average norm of the final residual vector, and number of function evaluations are used to compare the performance of the different scaling methods. In the control superblock, variables are similar in magnitude and the number of state variables is small. Hence the scaling methods have little impact on the convergence of the control superblock. Therefore, only the simulation for the airflow superblock and thermal superblock are discussed here. Results are presented for the entire 24-hour simulation period as well as results limited to the system occupied operation period (6AM – 6PM). Periods of time when the solver had difficulty finding a good solution are also discussed.

Overall, there are four termination conditions: 1) satisfying Eq. (13); 2) satisfying Eq. (14); 3) satisfying both Eqs. (13) and (14) simultaneously; and 4) failure to converge. Generally speaking, it is preferable that the residual test (14) is satisfied in order to ensure an accurate solution.

In this study, nonlinear equations resulting from the model are solved by the algorithm with varying degrees of success. Sometimes, only the less desirable convergence criterion, Eq. (13), is met. In this case, although the algorithm indicates successful convergence, the magnitude of the residual at the point of termination may still be large, and thus the solution does not solve, even approximately, the system of equations.

Table 4 summarizes the number of time steps for each test case that satisfies each of the first three convergence criteria in the summer 24-hour simulation period case. The column labelled (13) shows the number of times that only Eq. (13) was satisfied; the column labelled (14) shows the number of times that only Eq. (14) was satisfied; the column labelled (13) & (14) shows the number of times that both equations were satisfied. In the airflow superblock, there is little difference between the scaling cases. In the thermal superblock, the number of time steps that only satisfy Eq. (13) is greater for cases that employ uniform finite difference step length (cases 4, 5, 7 and 8). Further analysis of the convergence condition in the thermal superblock reveals that those time steps in which only Eq. (13) is satisfied, are mostly within the system unoccupied operation period.

Table 5 shows the number of time steps in which each of the three convergence criteria is satisfied during the summer occupied operation time period (6 AM to 6 PM). Comparing with Table 4, in the airflow superblock, the time steps that only satisfy Eq. (13) primarily appear during the operational period. In the thermal superblock, for cases 4, 5, 7 and 8, the number of time steps satisfying Eq. (13) drops greatly as compared to Table 4. However, these cases still satisfy only Eq. (13) more often than the other cases, which indicates worse performance.

To further study the accuracy of each case, the average norms of the final residuals are compared for each case and plotted in Figure 3 (for the airflow superblock) and Figure 4 (for the thermal superblock). In the airflow superblock (Figure 3), cases 2 through 5 (compared against case 1 with no scaling) show some improvement in the accuracy of the solver as they have smaller average norms of the final residual, however, the difference is not significant (with the greatest improvement, case 2, less than 20%).

Table 4 Termination Conditions of Airflow & Thermal Superblocks (Summer, 24-hour)

Superblock	Case	Convergence Condition		
		(13)	(14)	(13) & (14)
Airflow	1	405	10 953	23202
	2	425	11015	23120
	3	408	10977	23175
	4	406	10973	23181
	5	428	10944	23188
	6	402	10941	23217
	7	404	10938	23218
	8	402	10942	23216
Thermal	1	140	32457	1963
	2	91	33053	1416
	3	121	32618	1821
	4	5243	14778	14539
	5	12805	4584	17171
	6	137	32338	2085
	7	6187	15098	13275
	8	13064	4187	17309

Table 5 Termination Conditions of Airflow & Thermal Superblocks (Summer, 6AM-6PM)

Superblock	Case	Convergence Condition		
		(13)	(14)	(13) & (14)
Airflow	1	403	10921	5956
	2	423	10983	5874
	3	406	10945	5929
	4	404	10941	5935
	5	426	10912	5942
	6	400	10909	5971
	7	402	10906	5972
	8	400	10910	5970
Thermal	1	30	16146	1104
	2	31	16317	932
	3	27	16215	1038
	4	132	8693	8455
	5	468	3116	13696
	6	29	16099	1152
	7	200	8525	8555
	8	612	2784	13884

In the thermal superblock, shown in Figure 4, no case shows any improvement over case 1 (no scaling). The y-axis of this figure was scaled to focus on showing potential improvements in accuracy relative to case 1, so the results of cases 5, 7 and 8 for the whole day are greater than the maximum y-value. Their values are 1.39E-3, 9.39E-4 and 2.05E-3, respectively. It can be concluded that, on average, different scaling methods and finite difference step lengths do not improve accuracy (i.e., reduce residual) relative to the no scaling method in this superblock.

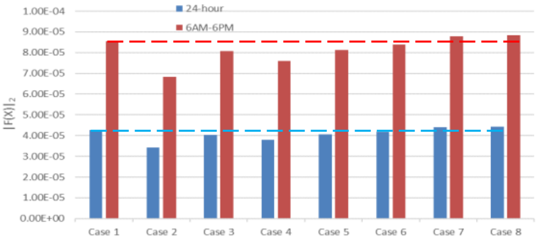

Figure 3 Average Norm of the Final Residual (Airflow Superblock, Summer)

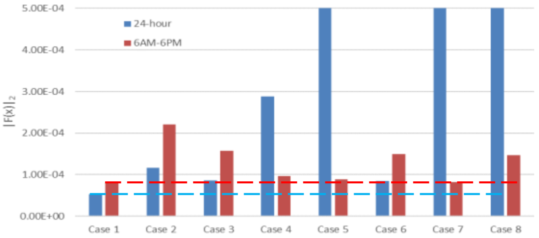

Figure 4 Average Norm of the Final Residual (Thermal Superblock, Summer)

The maximum and average number of function evaluations are compared for both the airflow and thermal superblocks in Table 6 and Table 7. A smaller number of function evaluations indicates less difficulty in the converging process and faster convergence speed. The results shown in these tables are consistent with those observed from the termination conditions analysis (Tables Table 4 and Table 5). That is, those test cases in which more time steps converge satisfying only Eq. (13), usually require more function evaluations.

Table 6 Number of Function Evaluations of Airflow & Thermal Superblocks (Summer, 24-hour)

	Airflow		Thermal	
Case	Max.	Ave.	Max.	Ave.
1	283	39.88	425	51.18
2	527	40.69	260	51.19
3	251	39.88	266	51.14
4	289	39.90	320	53.38
5	570	40.06	321	57.11
6	290	39.78	219	51.15
7	209	39.73	370	54.60
8	200	39.79	375	56.86

In the airflow superblock, to achieve the same level of accuracy, cases 6 through 8 require fewer function evaluations on average compared to the no scaling case (case 1, Table 6), but the difference is not significant. The maximum number of function evaluations for cases 3, 7 and 8 are less than for case 1. In the thermal superblock, cases that employ a uniform finite difference step length (case 4, 5, 7 and 8) require more function evaluations, especially during the unoccupied period. Other cases do not show much difference from case 1. However, all of the scaling cases, i.e., cases 2 through 8, have a smaller maximum number of function evaluations during the 24-hour simulation period relative to case 1 (no scaling).

This indicates better performance when it is difficult to reduce the residual.

Table 7 Number of Function Evaluations of Airflow & Thermal Superblocks (Summer, 6AM-6PM)

	Airflow		Thermal	
Case	**Max.**	**Ave.**	**Max.**	**Ave.**
1	283	40.75	425	51.24
2	527	42.37	260	51.24
3	245	40.73	266	51.20
4	289	40.78	320	51.71
5	570	41.10	321	52.17
6	290	40.55	219	51.17
7	209	40.45	370	51.77
8	200	40.57	375	52.11

In both the airflow and thermal superblocks, there exist some specific time periods that the norm of the final residual and number of function evaluations are much greater than other time periods, indicating that it is a greater challenge for the solver to converge during such periods. In the airflow superblock, the challenging periods are 8:47:00 AM - 8:47:45 AM, 9:54:00 AM - 10:04:00 AM, 12:06:00 PM - 12:08:00 PM, and 1:11:45 PM - 1:12:30 PM, denoted by ACP1 (Airflow Challenging Period 1), ACP2, ACP3 and ACP4, respectively. Variable scaling does not help reduce the norm of the final residuals during these time periods. However, the computational effort of the solver, i.e., the number of function evaluations, does differ when different scaling methods are applied. Table 8 shows the average number of function evaluations during these challenging periods. In general, cases that employ scaling method 3 (cases 6, 7 and 8) show better performance as they require fewer function evaluations, with the exception of case 8 in ACP4, which requires slightly more function evaluations than case 1 (no scaling). Other cases with better performance are case 3 in ACP1 and ACP2, and case 4 in ACP2.

Table 8 Average Number of Function Evaluations during Challenging Periods in the Airflow Superblock

Case	ACP1	ACP2	ACP3	ACP4
1	75.74	104.81	118.80	88.16
2	98.05	187.66	176.82	148.47
3	66.63	100.44	121.27	99.58
4	76.16	101.46	129.08	102.16
5	78.21	122.13	119.22	111.21
6	69.74	95.01	106.88	83.89
7	63.42	88.42	96.35	87.37
8	67.58	95.09	103.41	89.68

In the thermal superblock, there are three challenging periods: 6:11:00 AM - 6:12:30 AM, 10:21:30 AM - 10:23:15 AM, and 10:39:30 AM - 10:41:00 AM, denoted as TCP1 (Thermal Challenging Period 1), TCP2 and TCP3, respectively. Figure 5 shows a comparison of the norm of the final residuals during TCP2 between case 1 and case 2. Scaling does not show much improvement in reducing the residual in TCP1 and TCP3, while in TCP2, cases 2, 3, 4 and 7 successfully reduce the residual relative to case 1 (no scaling). Comparisons between cases 3, 4 and 7 and case 1 are similar to that shown in Figure 5, and thus they are not shown.

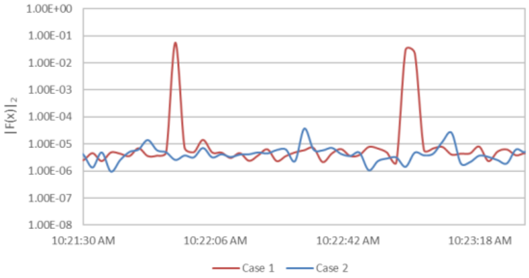

Figure 5 Norm of the Final Residual in the Thermal Superblock (10:21:30AM - 10:23:30AM)

The average number of function evaluations during challenging periods in the thermal superblock are shown in Table 9. Cases 3 and 6 demonstrate better performance as they require fewer function evaluations in all three challenging periods compared to case 1 (no scaling). In TCP2, all cases from 2 through 8 show improvement over case 1.

Table 9 Average Number of Function Evaluations during Challenging Periods in the Thermal Superblock

Case	TCP1	TCP2	TCP3
1	67.46	63.81	99.86
2	76.38	52.07	101.30
3	64.84	52.09	94.54
4	68.70	52.09	106.32
5	71.35	59.02	116.27
6	66.70	53.37	77.24
7	65.05	52.09	140.24
8	70.76	53.91	97.30

In general, most scaling cases demonstrate improvement in reducing the computing effort during the challenging periods, especially for cases 3, 6, 7 and 8. However, for both superblocks, residuals during challenging periods remain large, compared to the residual tolerance, no matter which scaling method is applied.

Further analysis in these periods shows that, in the airflow superblock, challenges usually appear when the VAV box dampers are near the minimum position. In spring and winter, the VAV box dampers are less dynamic, and the airflow superblock shows a high degree of success in convergence. Discontinuity of the model of the damper is very likely to be the cause of the convergence difficulties in this superblock.

In the thermal superblock, challenges arise when the air humidity ratio before the cooling coil is slightly larger than the air humidity ratio after the cooling coil, which indicates a partially wet coil condition. However, the current coil model only simulates either a completely dry or wet condition, making the coil model less stable and accurate under the partially wet condition. Discontinuity in the coil model is very likely to be the cause of the challenges in this superblock. The same phenomenon can be observed from the spring cases. In winter, since the cooling coil is not operating, the thermal superblock converges with a high degree of success. Future study is required to fully understand these phenomena.

Conclusion

The study demonstrates that scaling methods can have a strong impact on the computational efficiency of the solver in HVACSIM+. Among all tested scaling methods, case 6, which uses scaling method 3 with Dennis & Schnabel’s finite difference step-length, demonstrates the best performance as it reduces computational effort in all of the challenging periods and during the whole 24-hour simulation period. On the other hand, the accuracy of the converged solution is minimally impacted by differences in scaling. In the airflow superblock, because most variables have similar magnitudes of around $[10^{-2}, 10^{0}]$, scaling is not expected to have a major impact. However, it is surprising to find that the impact on the thermal superblock is small because the difference in magnitude between the temperature, $[10^{0}, 10^{2}]$ °C, and humidity variables, $[10^{-3}, 10^{-2}]$ kg/kg, is substantial (i.e., the temperature is as much as five orders or magnitude larger than the humidity ratio). One theory is that this system is too small and simple for the scaling to make much of an impact, so future work will include building a larger and more complex model to verify these findings. Challenges to convergence seem to be more related to the inaccurate and discontinuous nature of some of the component models, which may also mask the significance of scaling. More research is needed to identify the root causes of the challenging periods and to identify methods to improve the simulation accuracy during these periods.

Acknowledgement

This project is funded by National Institute of Standards and Technology under grant number: 60NANB15D251.

References

Dennis, J.E. and Schnabel, R.B. (1996) Numerical methods for unconstrained optimization and nonlinear equations, Siam.

DOE, US. (2011) Buildings energy data book, http://buildingsdatabook.eren.doe.gov/.

Fritzson, P. (2014) Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach, John Wiley & Sons.

Hiebert, K.L. (1982) An evaluation of mathematical software that solves systems of nonlinear equations, ACM Transactions on Mathematical Software (TOMS), 8, 5-20.

Klein, S., Beckman, W., Mitchell, J., Duffie, J., Duffie, N., Freeman, T., Mitchell, J., Braun, J., Evans, B. and Kummer, J. (2004) TRNSYS 16–A TRaNsient system simulation program, user manual, Solar Energy Laboratory. Madison: University of Wisconsin-Madison.

Park, C., Clark, D. and Kelly, G. (1986) HVACSIM+ building systems and equipment simulation program: building-loads calculation, National Bureau of Standards, Washington, DC (USA). Building Equipment Div.

Pernice, M. and Walker, H.F. (1998) NITSOL: A Newton iterative solver for nonlinear systems, SIAM Journal on Scientific Computing, 19, 302-318.

Pourarian, S., Kearsley, A., Wen, J. and Pertzborn, A. (2016) Efficient and robust optimization for building energy simulation, Energy and Buildings, 122, 53-62.

Powell, M.J. (1970) A hybrid method for nonlinear equations, Numerical methods for nonlinear algebraic equations, 7, 87-114.

Spark, L. (2003) SPARK 2.0 Reference Manual, Lawrence Berkeley National Laboratory and Ayres Sowell Associates Inc.

Wen, J. and Li, S. (2011) Tools for evaluating fault detection and diagnostic methods for air-handling units, ASHRAE RP-1312 Final Report, American Society of Heating, Refrigerating and Air Conditioning Engineers Inc., Atlanta.