2000-01-01

Improved Direct Solver for Building Energy Simulation

Michael Crowley
Technological University Dublin, michael.crowley@tudublin.ie

Saleem Hashmi
Dublin City University, saleem.hashmi@dcu.ie

Recommended Citation

Crowley, M., Hashmi, S.: Improved direct solver for building energy simulation. Building Services Engineering Research and Technology, Vol. 21, 3, pp.167-173. 2000. doi:10.1177/014362440002100304

Funder: Dublin Institute of Technology

Improved direct solver for building energy simulation

M.E. Crowley and M S J Hashmi

BUILDING SERV ENG RES TECHNOL 2000; 21; 167
DOI: 10.1177/014362440002100304

Proc. CIBSE A: Building Serv. Eng. Res. Technol. 21(3) 167–173 (2000) Printed in Great Britain B559

Summary Finite-difference methods, when applied to the differential equations modelling energy flows in buildings, give rise to a system of non-linear difference equations. A frequently used direct solution method involving linearisation is analysed and a related method is proposed. These and one other connected method are compared using a building-related test problem prepared for this purpose. The proposed numerical method is found to produce least error, about 30% less than the commonly used method, and it is recommended for inclusion in building energy simulation software. A fundamental method for estimating the pre-conditioning period of a building, arising from the work, is discussed.

Improved direct solver for building energy simulation

M E Crowley† BSc HDipEd CEng MCIBSE and **M S J Hashmi**‡ BSc MSc PhD DSc CEng FIMechE FIEI MASHME
†Department of Engineering Technology, Dublin Institute of Technology, Bolton Street, Dublin 1, Ireland
‡Head of School of Mechanical and Manufacturing Engineering, Dublin City University, Dublin 9, Ireland

Received 8 March 1999, in final form 30 May 2000

List of symbols

- c Specific heat of material represented by a node (J kg^{-1} K^{-1})
- d Slab thickness (m)
- $f(\ )$ Vector of derivative functions
- $g, \boldsymbol{G}$ Constituents of f
- h Space increment (m)
- i Space step level or node number
- $\boldsymbol{I}$ Identity matrix
- j Time step level
- $\boldsymbol{J}$ Jacobian matrix of f
- k Time increment (s)
- K Convergence factor
- L Lipschitz constant
- m Mass of material represented by a node (kg)
- n Total number of equations
- $O(\ )$ Order of magnitude
- $\boldsymbol{R}$ Alternative iteration function
- t Time (s)
- T Nodal temperature (K)
- $\boldsymbol{T}$ Vector of dependent variables
- x Space coordinate (m)
- $|\ |$ Magnitude
- $\|\ \|$ Spectral (l_2) norm

Greek symbols

- α Thermal diffusivity (m^2 s^{-1})
- δ Mean temperature difference between reference solution and test solution (K)
- $|\delta|$ Mean absolute temperature difference between reference solution and test solution (K)
- $|\hat{\delta}|$ Maximum absolute temperature difference between reference solution and test solution (K)
- λ_i Eigenvalues of $\boldsymbol{J}$
- μ_i Eigenvalues of Jacobian matrix of $\boldsymbol{R}$
- σ Stiffness ratio
- ϕ Nodal heat gain (W)

1 Introduction

In Europe and the United States over 50% of all energy use can be associated with buildings and a considerable portion of this is consumed to moderate internal environmental conditions[1]. The costs of this energy together with the global warming effect of the carbon dioxide produced by its conversion make energy-conscious design and operation of buildings imperative.

To this end a variety of building energy analysis and simulation tools are increasingly used to determine peak heating and cooling loads, to size thermal plant, to anticipate annual energy consumption and to analyse thermal comfort. These tools are generally produced as computer programs and range from coded manual methods to detailed 'first principles' energy models with few experimentally calibrated parameters. The mathematical techniques employed include response function methods, finite-difference methods and electrical analogue methods. Finite-difference techniques are considered the most flexible.

2 Model formulation and discretisation

A dynamic thermal model of a building consists of a set of partial differential equations (PDE) and ordinary differential equations (ODE) for the dependent temperatures and heat fluxes, which generally cannot be solved analytically. For example, the diffusion of heat through a solid building element, such as a homogeneous wall layer, is most often treated as one-dimensional and so the resulting temperature field can be described by the equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{1}$$

The finite-difference approach involves replacing the differential equations with consistent difference equations that are tractable. Solutions are obtained at discrete points in space and time rather than as continuous functions. One way of implementing this approach would be to decompose equation 1 into a set of ODEs by the method of lines[2], in which space is discretised but not time. A typical nodal equation would be

$$\frac{\mathrm{d}T_i}{\mathrm{d}t} = \frac{\alpha}{h^2}(T_{i-1} - 2T_i + T_{i+1}) \tag{2}$$

To these must be added ODEs for room air masses and other finite volumes of material assumed to have spatially uniform temperatures. Each volume is represented by a single nodal

temperature that varies in time according to an equation of the form

$$mc\frac{dT_i}{dt} = \sum\phi(t,T) \tag{3}$$

where the right-hand side represents the sum of the thermal driving forces acting on that node. The ϕ are in general non-linear functions of T. A complete building energy model can, therefore, be written succinctly as

$$\frac{\partial T}{\partial t} = f(t,T) \tag{4}$$

a vector equation depicting a non-linear system of first-order ODEs.

The above is, of necessity, a very brief description of a building thermal model. A detailed treatment of the construction of such a model is given by Clarke[1]. To complete the process of discretisation, numerical methods for ODEs are applied to equation 4. For instance, the first-order accurate Euler method gives the difference equation

$$T^{j+1} = T^j + kf(t^j, T^j) \tag{5}$$

and the second-order Trapezoidal Rule (equivalent to the Crank–Nicolson scheme for PDEs) gives

$$T^{j+1} = T^j + \frac{k}{2}\{f(t^j, T^j) + f(t^{j+1}, T^{j+1})\} \tag{6}$$

when applied to the same equation. The stability of any numerical method applied to equation 4 is determined by the value of the product $k\partial f/\partial T$ for a single equation and the products $k\lambda_i$ for a system of equations where the λ_i ($i = 1, 2, \ldots, n$) are the eigenvalues of $J = \partial f/\partial T$, the Jacobian matrix of f. For Euler's method, the product(s) must lie within a unit circle in the complex plane centred at (−1,0). The Trapezoidal Rule is described as A-stable because its region of stability is defined by $\mathrm{Re}(k\lambda)<0$, that is, the whole of the left half-plane.

The size of the time increment k is limited if Euler's method is applied to a building thermal model, the reason being that (4) is a *stiff* system of equations. An equation system, such as equation 4, is said to be stiff if

$$\mathrm{Re}(\lambda_i)<0 \tag{7}$$

and, in addition, the stiffness ratio σ satisfies

$$\sigma = \frac{\max_i|\mathrm{Re}(\lambda_i)|}{\min_i|\mathrm{Re}(\lambda_i)|} >> 0 \tag{8}$$

A stiff system is often referred to as one with a large Lipschitz constant L where

$$L = ||J|| \geq \max_i|\lambda_i| \geq \max_i|\mathrm{Re}(\lambda_i)| \tag{9}$$

When the physical entities or processes modelled by the equations have widely differing time constants ($1/|\mathrm{Re}(\lambda_i)|$), stiffness ensues. Systems may be considered marginally stiff if the stiffness ratio is $O(10)$, while ratios up to $O(10^6)$ are not uncommon. A building thermal model is moderately stiff. Sample rooms examined in connection with this work were found to have ratios of $O(10^2)$ for a lightweight room and $O(10^3)$ for a heavyweight room. Stiffness is not uncommon in practical problems arising in such fields as chemical kinetics, nuclear physics, process control, electronics and mathematical biology.

It can be seen that if Euler's method is applied to a stiff system $\max_i|\mathrm{Re}(\lambda_i)|$ forces a small time step k to keep all the products $k\lambda_i$ within the limited region of stability and a large number of such steps is required to pass through the slowest transient solution (with time constant $1/\min_i|\mathrm{Re}(\lambda_i)|$) to steady state. Euler's method is therefore computationally inefficient for stiff systems of equations, and this is the case for most explicit methods. The Trapezoidal Rule, being A-stable, is stable for all values of k but, of course, accuracy as well as stability must be considered when choosing a time increment.

Finally, in connection with time constants, it is worth drawing attention to a useful quantity that can be extracted from J. It is the *pre-conditioning period* of the building represented by equation 4. The pre-conditioning period is the simulation time required to allow the temperatures of all nodes to converge to values that are no longer affected by their arbitrarily chosen initial values. A number of different methods have been proposed to quantify it, including empirical relations and simulation experiments[3]. In this case an estimate is provided by calculating the time taken for the slowest transient solution of equation 4 (with time constant $1/\min_i|\mathrm{Re}(\lambda_i)|$) to decay to, say, 1% of its initial value. For the heavyweight example room mentioned above, the largest time constant was $1/(1.65\times10^{-6})$ implying a pre-conditioning period of 32.3 days. This is of the same order of magnitude as an estimate in Pinney and Parand[3] for a 'modern heavyweight' domestic building.

3 Solution of difference equations

The set of equations represented by (6) is implicit, requiring simultaneous solution at each time step. However, the additional work that this entails is often more than offset by a reduction in the number of steps needed. For instance, the time increment for Euler's method, an explicit method, must satisfy

$$\max_i|1 + k\lambda_i| < 1 \tag{10}$$

resulting typically in a limiting value for k of the order of minutes. Simulation runs ranging from a few days to a year are routinely undertaken.

As well as being implicit, equation 6 is non-linear and so an iterative solution method is indicated. The Newton–Raphson process is the most widely accepted method for stiff systems[2]. Applied to equation 6 it would take the form

$$T^{j+1} = T^{j+1} - \left\{I - \frac{k}{2}J(t^{j+1}, T^{j+1})\right\}^{-1} \times \left[T^{j+1} - T^j - \frac{k}{2}\{f(t^j, T^j) + f(t^{j+1}, T^{j+1})\}\right] \tag{11}$$

The Newton–Raphson method converges quadratically and generally it will converge for any time increment. However, a good initial estimate for T^{j+1} is required. A modified Newton–Raphson method is almost invariably employed in which triangular (LU) factorisation of the matrix $(I-kJ/2)$ replaces inversion and the same factors are used throughout the iteration. If the Jacobian does not vary too rapidly, it is often possible to retain the factors for a number of integration steps.

A simple fixed-point iteration can also be used in which equation 6 is iterated directly for T^{j+1}. The process is linearly convergent and will converge for any starting value, provided all the eigenvalues of the Jacobian matrix of the right-hand side are less than 1 in magnitude, in the neighbourhood of the solution. Differentiating the right-hand side of equation 6 with respect to each of the elements of T^{j+1} one gets $kJ(t^{j+1},T^{j+1})/2$, leading to the convergence condition

$$\frac{k}{2}\max_i|\lambda_i|<1 \tag{12}$$

Assuming equation 4 is stiff, this condition restricts the time increment to values similar to explicit methods; compare with condition (10). It is possible, however, to rearrange equation 6 so that a different iteration function appears on the right-hand side. It is shown in the appendix that, for this revised iterative method, stiffness actually promotes rapid convergence and long time steps are facilitated rather than prohibited.

In Clarke[1], for example, the function f in equation 6 is first decomposed in the manner done in equation 17. For instance, the $T_2^4-T_1^4$ expression in the long-wave radiation model is factorised to give

$$(T_2^2+T_1^2)(T_2+T_1)(T_2-T_1)$$

and the first two factors are included in G. There is no contribution to g from this expression. The terms are then rearranged to give equation 20, the alternative iterative method, which is renumbered and repeated here:

$$T^{j+1}=\left\{I-\frac{k}{2}G(t^{j+1},T^{j+1})\right\}^{-1}$$
$$\times\left[\left\{I+\frac{k}{2}G(t^j,T^j)\right\}T^j+\frac{k}{2}\{g(t^j,T^j)+g(t^{j+1},T^{j+1})\}\right] \tag{13}$$

Notice that the superscript notation of the appendix has been dropped and full arguments have been restored because the arguments in later equations may be evaluated at different time step levels.

4 Proposed method

Non-linear systems such as equation 6, when they crop up in building energy simulation[1] or more generally in conduction modelling[4], are usually linearised before being solved by matrix inversion or some equivalent direct process. Linearisation methods, such as extrapolation and lagging of dependent variables by one time step, eliminate the need for iteration. Equation 13 is linearised in Clarke[1] to give

$$T^{j+1}=\left\{I-\frac{k}{2}G(t^{j+1},T^j)\right\}^{-1}$$
$$\times\left[\left\{I+\frac{k}{2}G(t^j,T^{j-1})\right\}T^j+\frac{k}{2}\{g(t^j,T^{j-1})+g(t^{j+1},T^j)\}\right] \tag{14}$$

in which the dependent variables are evaluated one time step in arrears. All the terms on the right-hand side of equation 14 are known and so it can be solved directly for T^{j+1}.

Linearisation simplifies the solution of the problem, but there are some advantages in viewing the resulting direct solution process as the first iteration of an underlying iterative method:

- It is possible to investigate the benefits and costs of iterating more than once. Generally, stiffer systems require fewer iterations to achieve a given level of accuracy.
- The convergence factor K can be estimated using equation 21.
- A number of apparently different direct methods can be produced by changing the initial estimate used and iterating just once.

Regarding the final point, it is necessary to estimate T^{j+1}, the unknown, on the right-hand side of equation 13 before iteration can commence. Equation 14 is generated by substituting for both T^{j+1} and T^j in equation 13 even though the latter is already known. If this unnecessary substitution is eliminated, another direct method can be put forward:

$$T^{j+1}=\left\{I-\frac{k}{2}G(t^{j+1},T^j)\right\}^{-1}$$
$$\times\left[\left\{I+\frac{k}{2}G(t^j,T^j)\right\}T^j+\frac{k}{2}\{g(t^j,T^j)+g(t^{j+1},T^j)\}\right] \tag{15}$$

Notice that the proposed method requires just one vector of starting values whereas equation 14 requires two and so must be primed using an independent single-step method. One further initial estimate can be formed by using a Newton–Gregory extrapolation from previous time steps. T^{j+1} in equation 13 is replaced by $\hat{T}^{j+1}=2T^j-T^{j-1}$, leading to the method:

$$T^{j+1}=\left\{I-\frac{k}{2}G(t^{j+1},\hat{T}^{j+1})\right\}^{-1}$$
$$\times\left[\left\{I+\frac{k}{2}G(t^j,T^j)\right\}T^j+\frac{k}{2}\{g(t^j,T^j)+g(t^{j+1},\hat{T}^{j+1})\}\right] \tag{16}$$

5 Evaluation of numerical methods

Three direct solution methods, arising from the iterative method specified by equation 13, are available for assessment. They will be referred to as LL (linearisation by lagging, equation 14), PM (the proposed method, equation 15) and LE (linearisation by extrapolation, equation 16). The Newton–Raphson (NR) method was also included for comparison because it is so widely used, together with the Trapezoidal Rule, to solve stiff systems in a wider context. Each method can, of course, be iterated to convergence, but few practical building energy applications require such rigour[1].

5.1 Test problem

Analytical tests, based on physically simple heat transfer problems with known solutions, are decisive but very limited in scope. Empirical validation using measured data from a real structure, a necessary and appropriate application of the scientific method to whole model validation, is unsuitable here because it is difficult to separate the error due to the numerical method, which is sought, from errors in other parts of the model and in the input data. A mathematical test was used in

which the methods were applied to an equation set with the attributes of the building energy problem that is characterised mathematically in Crowley and Hashmi[5]. The test equations were generated by considering the heat flows at a cubic space enclosed by five identical plane slabs and one vertical glass sheet. Each 3 m-square slab was represented by three nodes and exchanged heat by convection with the enclosed air mass, as did the glass sheet, which was represented by one node. Internal long-wave radiation was exchanged between opposite faces only. External surfaces were exposed to a sinusoidally varying air temperature with a period of 24 h, and no other thermal influence. Short-wave radiation, entering through the glass, acted on just one internal surface. This solar term was represented by the positive part of a sinusoid with a 15% ripple superimposed. A casual heat gain to the internal air mass was switched on in the morning and off again in the afternoon. A proportionally controlled convective air-conditioning terminal unit could be activated for the whole of the simulated period.

This test example is small enough to be computed quickly and yet detailed enough to capture the essential features of the application. It is a demanding problem that includes step changes and discontinuous derivatives in the thermal driving terms. It consists of 17 differential equations that are, in general, non-linear, and stiffness ratios ranging from $O(10)$ to $O(10^4)$ were generated during the testing process.

5.2 *Computational procedures*

Programs were produced for the four numerical methods being assessed. Each incorporated one iteration and one matrix inversion per time step. A Newton–Gregory extrapolation was used to construct a starting value for the Newton–Raphson method.

The work was carried out on a personal computer using a general-purpose mathematical software package[6]. During a typical test run, two independent solutions were generated using built-in differential equation solvers, and a reference solution was formed by averaging them. Both of these methods, the method of Rosenbrock and the fourth-order Runge–Kutta method[6, 7], include adaptive step-size control, and the tolerance variable was set to 10^{-6} in each case. The agreement between these two solutions was excellent (Table 1). The four test solutions were calculated at 15 minute and 1 hour intervals. The longer time increment led to large errors in the test solutions when the terminal unit was inactive (Table 1). Further iteration reduced these errors, but they were still appreciable, indicating the need for shorter time steps, at least where thermal disturbance was most intense. It was decided to carry out the assessment using a time increment of 15 minutes. The reference solution was subtracted from each of the test solutions in turn at every node and time step over a 4 day period following the pre-conditioning period. The statistics presented in Table 1 were extracted from the sets of differences for two test runs. The cross-correlation coefficient gives a measure of the phase relationship between the reference solution and each of the other solutions.

Test runs were carried out using slabs of the first four materials listed in Table 2, which between them virtually span the range of thermal diffusivities encountered in building materials. A variety of slab thicknesses was used leading to characteristic conduction times, d^2/α, ranging from 1 s to 26 days and a correspondingly large range of stiffness ratios. Discontinuities in the heat gains were expected to lead to the greatest thermal disturbance, so tests were carried out with both the step changes and the discontinuous derivatives occurring a fixed amount of time before some of the assessment points. Time delays (prior to assessment) of between 2 and 8 minutes were used, the shortest time constant for 0.1 m concrete construction being 5 minutes in the absence of the terminal unit and less than 1 minute with the unit active. The casual heat gain period was also moved back and then forward by 1 hour so as to substantially change its time of application relative to other loads. These changes in timing were examined lest fixed relative times favoured some numerical methods. In all cases tests were done with the free-running cell, and then repeated with the terminal unit active and sized for 120% of the peak thermal load. A 2 K proportional band was used.

Table 1 Accuracy statistics for test runs 1 and 2

Numerical method	Terminal unit status	Time increment (s)	Temperature difference between reference solution and other solutions (K)			Cross-correlation at zero time delay (air point node only)
			Mean difference δ	Mean absolute difference $\lvert\delta\rvert$	Maximum absolute difference $\lvert\hat{\delta}\rvert$	
Rosenbrock	On	Variable	-9.10×10^{-8}	4.53×10^{-7}	1.38×10^{-5}	1.0000
	Off	Variable	2.71×10^{-7}	6.50×10^{-7}	2.19×10^{-4}	1.0000
Runge-Kutta	On	Variable	9.10×10^{-8}	4.53×10^{-7}	1.38×10^{-5}	1.0000
	Off	Variable	-2.71×10^{-7}	6.50×10^{-7}	2.19×10^{-4}	1.0000
LL	On	900	2.36×10^{-5}	1.46×10^{-3}	4.41×10^{-2}	1.0000
		3600	-5.58×10^{-4}	6.66×10^{-3}	1.80×10^{-1}	0.9999
	Off	900	6.95×10^{-4}	6.18×10^{-3}	1.41	1.0000
		3600	5.74×10^{-3}	4.93×10^{-2}	4.44	0.9997
PM	On	900	1.40×10^{-5}	8.77×10^{-4}	2.70×10^{-2}	1.0000
		3600	-8.20×10^{-4}	6.05×10^{-3}	9.09×10^{-2}	0.9999
	Off	900	4.48×10^{-4}	5.46×10^{-3}	1.41	1.0000
		3600	1.88×10^{-3}	1.92×10^{-1}	7.26	0.9996
LE	On	900	1.25×10^{-5}	6.52×10^{-4}	2.53×10^{-2}	1.0000
		3600	-4.19×10^{-4}	4.57×10^{-2}	7.69×10^{-1}	0.9999
	Off	900	3.54×10^{-4}	7.59×10^{-3}	1.35	1.0000
		3600	-3.80×10^{-3}	9.28×10^{-2}	3.75	0.9997
NR	On	900	2.66×10^{-5}	7.29×10^{-4}	4.25×10^{-2}	1.0000
		3600	-9.06×10^{-4}	5.70×10^{-3}	6.39×10^{-2}	0.9999
	Off	900	3.11×10^{-4}	4.91×10^{-3}	9.18×10^{-1}	1.0000
		3600	7.85×10^{-4}	3.45×10^{-2}	2.59	0.9998

Table 2 Material properties

	Thickness (m)	Conductivity ($W m^{-1}K^{-1}$)	Density ($kg m^{-3}$)	Specific heat ($J kg^{-1}K^{-1}$)	Thermal diffusivity ($m^2 s^{-1}$)
Aluminium	0.002	200	2800	880	81.17×10^{-6}
Insulation	0.10	0.045	50	840	1.07×10^{-6}
Concrete	0.20	1.9	2300	840	0.98×10^{-6}
Wood	0.10	0.14	500	2500	0.11×10^{-6}
Glass	0.005	1.05[a]	2500	750	0.56×10^{-6}

[a] Not utilized

5.3 *Comparison of methods*

The results obtained for the test runs outlined in section 5.2 are given in Table 3. The difference statistics for LL were divided into the corresponding statistics for each of the other three methods in turn. The geometric mean values of these ratios, calculated for the full set of test runs in each case, are presented in Table 4. They measure the average factor by which the difference statistic is changed when LL is replaced by one of the other methods.

All three methods achieved a reduction in both mean absolute difference and maximum absolute difference when compared with LL. The decrease in maximum error was greatest for PM at 27%. Mean error reduction was greatest for NR, but this method requires the construction of a Jacobian matrix or, at least, an approximation to it. PM achieved a reduction in mean error of 29%. The performance of LE was not as good as might be expected. Its initial estimate is extrapolated from two previous solution values and should, therefore, be better than the one used with the proposed method. LL performed as expected. Its initial estimate is the same as that used with PM, namely the last solution value, but an unnecessary substitution is also made, which reduces the accuracy of the method.

6 Conclusions

The use of finite-difference methods to discretise the differential equations representing heat flows in buildings and elsewhere produces a system of algebraic equations that are, in general, non-linear. A commonly used linearisation procedure results in a direct solution method that can be regarded as the first iteration of an underlying iterative method. The iterative process is examined and found to be well suited to the solution of stiff systems. Two other direct methods emerge from this iterative procedure. One is a proposed change to the previously mentioned linearisation scheme and the other involves extrapolation. The proposed method was found to be the most accurate of the three direct solution methods for a representative test problem. Generally similar results were obtained for three other test examples, two of which were building related. The improved accuracy can, of course, be traded for greater speed of execution. The proposed method is a single-step one requiring only minor changes in building energy simulation software that includes the more commonly used linearisation method.

All of the tested methods can, of course, be used in conjunction with other implicit solvers for ODEs, for example the Backward Differentiation Formulae. Another of these implicit methods has been shown[5] to offer significant accuracy and stability advantages over the Trapezoidal Rule in this application.

Appendix

Another iterative solution method for equation 6 can be constructed if the function f can be decomposed in the following way:

$$f(t, T) = G(t, T)T + g(t, T) \tag{17}$$

where $|G|$ is large and $|\partial g/\partial T|$ is small over the interval of interest; that is, the stiffness of f expresses itself in G rather than in g. In this case equation 6 becomes

$$T^{j+1} = T^j + \frac{k}{2}(G^jT^j + g^j + G^{j+1}T^{j+1} + g^{j+1}) \tag{18}$$

Here superscripts have been placed on the functions to indicate the time step level. Equation 18 can be rearranged to give

$$\left(I - \frac{k}{2}G^{j+1}\right)T^{j+1} = \left(I + \frac{k}{2}G^j\right)T^j + \frac{k}{2}(g^j + g^{j+1}) \tag{19}$$

Therefore

$$T^{j+1} = \left(I - \frac{k}{2}G^{j+1}\right)^{-1}\left\{\left(I + \frac{k}{2}G^j\right)T^j + \frac{k}{2}(g^j + g^{j+1})\right\} \tag{20}$$

Equation 20 is another possible fixed-point iteration. If its right-hand side is denoted by R and if $\mu_i\ (i = 1, 2, \ldots, n)$ are the eigenvalues of $\partial R/\partial T^{j+1}$, the Jacobian matrix of R, the condition for convergence of equation 20 is

$$\max_i|\mu_i| = K < 1 \tag{21}$$

and small values for K result in rapid convergence. A typical column of the Jacobian would be

$$\frac{\partial R}{\partial T_i^{j+1}} = \left(I - \frac{k}{2}G^{j+1}\right)^{-1}\left(\frac{k}{2}\frac{\partial g^{j+1}}{\partial T_i^{j+1}}\right) + \left\{-\left(I - \frac{k}{2}G^{j+1}\right)^{-1}\left(-\frac{k}{2}\frac{\partial G^{j+1}}{\partial T_i^{j+1}}\right)\left(I - \frac{k}{2}G^{j+1}\right)^{-1}\right\} \times \left\{\left(I + \frac{k}{2}G^j\right)T^j + \frac{k}{2}(g^j + g^{j+1})\right\} \tag{22}$$

The identity

$$\frac{\partial(M^{-1})}{\partial T} \equiv -M^{-1}\frac{\partial M}{\partial T}M^{-1} \tag{23}$$

Table 3 Accuracy achieved for the test problem

Test run	Test space construction	Slab thickness d (m)	Characteristic conduction time d^2/α (s)	Average stiffness ratio	Terminal unit status	Time delay prior to assessment α (s)	Displacement of casual heat gain (s)	Numerical method							
								LL		PM		LE		NR	
								$\lvert\delta\rvert$	$\lvert\hat{\delta}\rvert$	$\lvert\delta\rvert$	$\lvert\hat{\delta}\rvert$	$\lvert\delta\rvert$	$\lvert\hat{\delta}\rvert$	$\lvert\delta\rvert$	$\lvert\hat{\delta}\rvert$
1	Concrete	0.100	1.02×10^4	309	On	180	0	1.46×10^{-3}	4.41×10^{-2}	8.77×10^{-4}	2.70×10^{-2}	6.52×10^{-4}	2.35×10^{-2}	7.29×10^{-4}	4.25×10^{-2}
2	Concrete	0.100	1.02×10^4	49	Off	180	0	6.18×10^{-3}	1.41	5.46×10^{-3}	1.41	7.59×10^{-3}	1.35	4.91×10^{-3}	9.18×10^{-1}
3	Insulation	0.100	9.35×10^3	57	On	180	0	5.02×10^{-3}	1.27×10^{-1}	3.07×10^{-3}	9.21×10^{-2}	8.90×10^{-3}	3.87×10^{-1}	1.35×10^{-3}	5.11×10^{-2}
4	Insulation	0.100	9.35×10^3	49	Off	180	0	3.67×10^{-2}	1.47	3.10×10^{-2}	1.47	7.79×10^{-2}	1.55	2.52×10^{-2}	1.14
5	Wood	0.100	9.09×10^4	700	On	180	0	2.38×10^{-3}	4.09×10^{-2}	1.39×10^{-3}	2.89×10^{-2}	9.43×10^{-4}	2.51×10^{-2}	9.22×10^{-4}	3.90×10^{-2}
6	Wood	0.100	9.09×10^4	122	Off	180	0	1.08×10^{-2}	1.34	1.01×10^{-2}	1.34	1.51×10^{-2}	1.28	9.32×10^{-3}	8.51×10^{-1}
7	Aluminium	0.100	1.23×10^2	1769	On	180	0	1.02×10^{-3}	4.91×10^{-2}	6.70×10^{-4}	2.81×10^{-2}	5.53×10^{-4}	2.19×10^{-2}	6.50×10^{-4}	4.20×10^{-2}
8	Aluminium	0.100	1.23×10^2	1881	Off	180	0	5.66×10^{-3}	1.65	4.50×10^{-3}	1.65	2.01×10^{-2}	3.04	4.26×10^{-3}	1.15
9	Concrete	0.050	2.55×10^3	146	On	180	0	1.89×10^{-3}	3.96×10^{-2}	1.19×10^{-3}	2.66×10^{-2}	7.57×10^{-4}	4.33×10^{-2}	7.96×10^{-4}	4.03×10^{-4}
10	Concrete	0.050	2.55×10^3	43	Off	180	0	6.79×10^{-3}	1.40	5.97×10^{-3}	1.40	9.55×10^{-3}	1.35	5.38×10^{-3}	9.18×10^{-1}
11	Concrete	0.200	4.08×10^4	690	On	180	0	9.25×10^{-4}	5.22×10^{-2}	7.45×10^{-4}	4.49×10^{-2}	7.92×10^{-4}	3.55×10^{-2}	7.44×10^{-4}	6.06×10^{-2}
12	Concrete	0.200	4.08×10^4	120	Off	180	0	5.97×10^{-3}	1.61	4.86×10^{-3}	1.61	1.02×10^{-2}	1.51	4.44×10^{-3}	1.08
13	Concrete	0.100	1.02×10^4	309	On	180	−3600	1.46×10^{-3}	4.56×10^{-2}	8.42×10^{-4}	2.60×10^{-2}	6.38×10^{-4}	2.21×10^{-2}	7.28×10^{-4}	4.05×10^{-2}
14	Concrete	0.100	1.02×10^4	49	Off	180	−3600	6.76×10^{-3}	1.72	5.53×10^{-3}	1.72	1.35×10^{-2}	1.61	5.26×10^{-3}	1.18
15	Concrete	0.100	1.02×10^4	310	On	180	+3600	1.45×10^{-3}	4.35×10^{-2}	9.04×10^{-4}	2.64×10^{-2}	6.08×10^{-4}	2.36×10^{-2}	6.86×10^{-4}	4.07×10^{-2}
16	Concrete	0.100	1.02×10^4	49	Off	180	+3600	6.20×10^{-3}	1.25	5.64×10^{-3}	1.25	5.49×10^{-3}	1.21	5.01×10^{-3}	7.83×10^{-1}
17	Concrete	0.100	1.02×10^4	309	On	120	0	1.48×10^{-3}	4.43×10^{-2}	9.41×10^{-4}	2.71×10^{-2}	7.07×10^{-4}	1.48×10^{-2}	7.66×10^{-4}	4.26×10^{-2}
18	Concrete	0.100	1.02×10^4	49	Off	120	0	7.80×10^{-3}	1.94	7.05×10^{-3}	1.94	9.10×10^{-3}	1.87	6.57×10^{-3}	1.44
19	Concrete	0.100	1.02×10^4	309	On	240	0	1.43×10^{-3}	4.40×10^{-2}	8.22×10^{-4}	3.15×10^{-2}	5.97×10^{-4}	3.17×10^{-2}	6.82×10^{-4}	4.63×10^{-2}
20	Concrete	0.100	1.02×10^4	49	Off	240	0	5.06×10^{-3}	1.25	4.30×10^{-3}	9.88×10^{-1}	6.34×10^{-3}	9.24×10^{-1}	3.84×10^{-3}	5.03×10^{-1}
21	Concrete	0.100	1.02×10^4	309	On	300	0	1.43×10^{-3}	4.39×10^{-2}	7.91×10^{-4}	3.30×10^{-2}	5.68×10^{-4}	3.33×10^{-2}	6.56×10^{-4}	4.80×10^{-2}
22	Concrete	0.100	1.02×10^4	49	Off	300	0	3.88×10^{-3}	1.24	3.08×10^{-3}	6.51×10^{-1}	5.20×10^{-3}	5.88×10^{-1}	2.96×10^{-3}	4.80×10^{-1}
23	Concrete	0.100	1.02×10^4	309	On	360	0	1.44×10^{-3}	4.38×10^{-2}	7.87×10^{-4}	3.36×10^{-2}	5.63×10^{-4}	3.38×10^{-2}	6.52×10^{-4}	4.86×10^{-2}
24	Concrete	0.100	1.02×10^4	49	Off	360	0	3.20×10^{-3}	1.23	2.17×10^{-3}	4.63×10^{-1}	4.39×10^{-3}	3.23×10^{-1}	2.30×10^{-3}	5.58×10^{-1}
25	Concrete	0.100	1.02×10^4	309	On	420	0	1.45×10^{-3}	4.37×10^{-2}	8.08×10^{-4}	3.40×10^{-2}	5.80×10^{-4}	3.40×10^{-2}	6.94×10^{-4}	4.88×10^{-2}
26	Concrete	0.100	1.02×10^4	49	Off	420	0	3.17×10^{-3}	1.21	1.84×10^{-3}	4.45×10^{-1}	3.82×10^{-3}	4.84×10^{-1}	2.06×10^{-3}	7.49×10^{-1}
27	Concrete	0.100	1.02×10^4	309	On	480	0	1.48×10^{-3}	4.36×10^{-2}	8.45×10^{-4}	3.43×10^{-2}	6.25×10^{-4}	3.42×10^{-2}	7.70×10^{-4}	4.90×10^{-2}
28	Concrete	0.100	1.02×10^4	49	Off	480	0	3.62×10^{-3}	1.20	2.30×10^{-3}	5.06×10^{-1}	4.24×10^{-3}	6.37×10^{-1}	3.47×10^{-3}	9.04×10^{-1}
29	Wood	0.500	2.27×10^6	9311	On	180	0	1.37×10^{-3}	3.95×10^{-2}	1.00×10^{-3}	3.42×10^{-2}	7.54×10^{-4}	3.41×10^{-2}	7.98×10^{-4}	4.61×10^{-2}
30	Wood	0.500	2.27×10^6	1745	Off	180	0	5.59×10^{-3}	1.07	5.75×10^{-3}	9.99×10^{-1}	8.28×10^{-3}	1.05	3.83×10^{-3}	6.21×10^{-1}
31	Aluminium	0.010	1.23	17530	On	180	0	2.01×10^{-3}	6.83×10^{-2}	1.28×10^{-3}	2.80×10^{-2}	9.61×10^{-4}	4.44×10^{-2}	8.74×10^{-4}	3.94×10^{-2}
32	Aluminium	0.010	1.23	19580	Off	180	0	6.60×10^{-3}	1.54	5.52×10^{-3}	1.54	2.45×10^{-2}	2.83	5.31×10^{-3}	1.08

Table 4 Geometric mean reduction in error achieved for the test problem when LL is replaced by other numerical methods

Accuracy statistic	Numerical method			
	LL	PM	LE	NR
$\lvert\delta\rvert$	1.000	0.709	0.880	0.610
$\lvert\hat{\delta}\rvert$	1.000	0.731	0.769	0.757

has been used here. The derivative of an array $\boldsymbol{M}$ (matrix or vector) is defined as an array whose elements are the derivatives of the elements of $\boldsymbol{M}$.

Substituting from equation 19 for the final expression in braces, equation 22 simplifies to

$$\frac{\partial \boldsymbol{R}}{\partial T_i^{j+1}} = \left(\boldsymbol{I} - \frac{k}{2}\boldsymbol{G}^{j+1}\right)^{-1}\frac{k}{2}\left(\frac{\partial \boldsymbol{G}^{j+1}}{\partial T_i^{j+1}}\boldsymbol{T}^{j+1} + \frac{\partial \boldsymbol{g}^{j+1}}{\partial T_i^{j+1}}\right)\Bigg\} \tag{24}$$

The properties of the alternative iterative method can be deduced from this last equation. Most importantly, $\boldsymbol{G}$ appears only in the inverted expression and, because $\|\boldsymbol{G}\|$ is assumed to be large (and found to be so for the systems examined), $\partial \boldsymbol{R}/\partial \boldsymbol{T}^{j+1}$ will have small elements and so $\|\partial \boldsymbol{R}/\partial \boldsymbol{T}^{j+1}\|$ will be small. Stiffness, therefore, increases the rate of convergence since

$$K = \max_i|\mu_i| \leqslant \|\partial \boldsymbol{R}/\partial \boldsymbol{T}^{j+1}\|$$

It can also be inferred from equation 24 that K increases only very slowly with k when the $k\boldsymbol{G}^{j+1}/2$ term dominates the inverted expression. In other words, large time increments do not jeopardise convergence. The convergence rate of equation 20 is, of course, also dependent on the magnitudes of $\boldsymbol{T}^{j+1}$, $\partial \boldsymbol{g}^{j+1}/\partial T_i^{j+1}$ and $\partial \boldsymbol{G}^{j+1}/\partial T_i^{j+1}$ as measured by their norms.

Finally, it is worth noting that the three iterative methods:

— simple fixed point iteration (equation 6)

— the alternative iterative method (equation 20) and

— the Newton–Raphson process (equation 11)

become one when $\|\boldsymbol{J}\|$, and consequently $\|\boldsymbol{G}\|$, is small.

References

1 Clarke J A *Energy simulation in building design* (Bristol: Adam Hilger) (1985)

2 Lambert J D *Numerical methods for ordinary differential systems: the initial value problem* (Chichester: Wiley) (1991)

3 Pinney A and Parand F The effect of computational parameters on the accuracy of results from detailed thermal simulation programs *Proceedings of Building Environmental Performance* (University of Kent, Canterbury: BEPAC) pp 207–222 (Apr 1991)

4 Ozisik M N *Heat conduction* 2nd edition (New York: Wiley) (1993)

5 Crowley M E and Hashmi M S J Evaluation of implicit numerical methods for building energy simulation *Proc. Inst. Mech. Eng., Part A, Journal of Power and Energy* 212(A5) 331–342 (1998)

6 *User's guide, Mathcad PLUS 6.0* (Cambridge, MA: Mathsoft Inc) (1995)

7 Press W H, Flannery B P, Teukolsky S A and Vetterling W T *Numerical recipes in C, the art of scientific computing* 2nd edition (Cambridge: Cambridge University Press) (1992)