Investigation and Development of Implicit Numerical Methods for Building Energy Simulation

Michael Edward Crowley BSc, HDipEd, CEng, MCIBSE

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Supervisor: Professor M.S.J. Hashmi
Head of School of Mechanical and Manufacturing Engineering
Dublin City University

September 2005

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Michael Crowley ID No: 95971505

Date: 27 Sept. 2005

Contents

Abstract

Investigation and Development of Implicit Numerical Methods for Building Energy Simulation

Michael E. Crowley BSc, HDipEd, CEng, MCIBSE

A variety of building energy analysis and simulation tools are increasingly used to determine peak heating and cooling loads, size thermal plant, anticipate annual energy consumption and analyse thermal comfort. Numerical solution techniques are considered the most flexible for building energy simulation. When applied to the differential equations modelling energy flows in buildings, they give rise to a system of non-linear algebraic (difference) equations.

In order to evaluate numerical methods for building energy simulation, the problem has been characterized mathematically and comprehensive test problems (equation sets) with these characteristics have been prepared. The principal attribute of the problem was found to be a stiffness ratio of the order of 10^4. Candidate methods have been programmed and their outputs compared, in *numerical experiments*, with highly accurate (converged) solutions for the test problems. The accepted validation methods, empirical validation, analytical verification and inter-modal comparison were considered inappropriate. The first estimates total and not just numerical error, the second is too confined and the third lacks an absolute standard. The main evaluation parameter used was computational efficiency which is defined as accuracy attained per unit (computational) effort expended.

An improved difference equation solver has been proposed and compared with the one used in the European reference model (ESP) and elsewhere. It was found to produce 27% less error than the currently used method. A fundamental method for estimating the pre-conditioning period of a building has been put forward in this part of the work. The trapezoidal rule (TR) is currently used in a number of building energy simulation packages including ESP. A known instability associated with the method is described and an implicit member of the Runge-Kutta family, possessing the necessary strong stability, has been shown, using the test problems, to be more efficient than TR by a factor of 4.27.

List of Figures

List of Tables

Acknowledgements

Firstly, my gratitude is sincerely expressed to my supervisor, Professor Saleem Hashmi, for his support and patience throughout this period during which he always found time for advice, guidance and discussion.

My thanks are also due to numerous colleagues and fellow postgraduate students for the many stimulating debates on aspects of the work – in particular to Louis Demetriou and Ben Costelloe.

I would like to thank the very professional and helpful staff at the libraries of Dublin City University and the Dublin Institute of Technology – my 'gateways'.

I am indebted to the Dublin Institute of Technology for the fee support and research time provided.

Finally, I am grateful to the three generations of my family for their support and tolerance over the period. I dedicate this dissertation to them.

Notation

Bi	Biot number, $h_c d_{1/2}/k_s$ (dimensionless)
Bi_{fd}	finite-difference form of the Biot number, $h_c h/k_s$ (dimensionless)
c	specific heat of material represented by a node (J /kg K)
C_{lte}	local truncation error constant (dimensionless)
d	slab thickness (m)
$d_{1/2}$	slab half-thickness (m)
$e(\cdot)$	thermal excitation
ET	execution time
$f_i(\cdot)$	derivative function
$f_i'(\cdot)$	time derivative of f_i
$\mathbf{f}(\cdot)$	vector of derivative functions
Fo	Fourier number, $\alpha t/d_{1/2}{}^2$ (dimensionless)
Fo_{fd}	mesh ratio or finite-difference form of the Fourier number, $\alpha k/h^2$ (dimensionless)
$\mathbf{g}, \mathbf{G}$	constituents of $\mathbf{f}$
h	space increment (m)
h_c	convection coefficient (W /m^2 K)
h_{is}	inside surface convection coefficient (W /m^2 K)
h_{os}	outside surface convection coefficient (W /m^2 K)
i	space step level, node number
$\mathbf{I}$	identity matrix
j	time step level
$\mathbf{J}$	Jacobian matrix of $\mathbf{f}$
k	time increment (s)
k_s	conductivity of slab (W /m K)
K	convergence factor
l	linear dimension (m)
L	Lipschitz constant
$L(\cdot)$	Laplace transform

m	mass of material represented by a node (kg)
MI	number of matrix inversions carried out during a test run
n	total number of equations or nodes
$\mathbf{N}$	Newton iteration matrix
N	number of machine operations per node/equation for a test run
$r(\cdot)$	overall response function, amplification factor
r', r''	successive time derivatives of r
$\mathbf{R}$	alternative iteration function
s	number of Runge-Kutta stages
S	stiffness ratio
t	time (s)
t^*	dimensionless time
T	nodal temperature (K)
T_a	air temperature (K)
T_{in}	initial slab temperature (K)
T^*	dimensionless temperature
T', T'', T'''	successive time derivatives of T
$\mathbf{T}$	vector of dependent variables
$\mathbf{T}'$	time derivative of $\mathbf{T}$
$u(\cdot)$	unit response function
w	complex number, $k\lambda$
x	space co-ordinate (m)
x^*	dimensionless space co-ordinate
z	exponent
$\lvert\cdot\rvert$	magnitude or modulus
$\lVert\cdot\rVert$	spectral (l_2) norm

Greek symbols

α	thermal diffusivity (m^2 /s); parameter (dimensionless)
γ	weighting factor (dimensionless); parameter (dimensionless)
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)

ε round-off error in dependent variable

ζ fraction of time step (dimensionless)

$\theta(\cdot)$ temperature distribution function

λ complex number

λ_i eigenvalues of $\mathbf{J}$

μ_i eigenvalues of Jacobian matrix of $\mathbf{R}$

ρ density (kg/m^3)

σ Stefan-Boltzmann constant

τ characteristic time scale of a thermal disturbance (s)

$\tau_{\min}$ characteristic time scale of the most dynamic thermal disturbance (s)

ϕ nodal heat gain (W)

Abbreviations

Alex2 Alexander's second-order method

BDF backward differentiation formulae

BDF2 second-order backward differentiation formula

BEM backward Euler method

BFD4 fourth-order backward-forward difference method

CE computational efficiency

DIRK diagonally implicit Runge-Kutta method

ER Euler's rule

ESDIRK singly diagonally implicit Runge-Kutta method with an explicit first stage

FSAL first-same-as-last

IIE implicit improved Euler

IRK implicit Runge-Kutta method

Kvaerno3 Kvaerno's third-order method

LE linearization by extrapolation

LL linearization by lagging

LMM linear multi-step method

LTE local truncation error

MAT mean access time

MOLCOL modified one-leg collocation method

NR Newton–Raphson method

$O(\cdot)$ order of magnitude

ODE ordinary differential equation

PC predictor-corrector

PDE partial differential equation

PM proposed method

RAM random access memory

$\mathrm{Re}(\cdot)$ real part of a complex number

RK2 second-order Runge-Kutta method

ROS2 the method of Verwer *et al.*

SDIRK singly diagonally implicit Runge-Kutta method

SM Scraton's method

TR trapezoidal rule

TR-BDF2 trapezoidal rule – backward differentiation formula composite method

TRX2 trapezoidal rule by two

VSVO variable step, variable order

Chapter 1:

Introduction

1.1 Energy use in buildings

Ever since the first world energy crisis of 1973 there has been increasing scrutiny of energy use with a view to reducing reliance on non-renewable energy sources. The cost and availability of fossil fuels are always of concern, even more so now with the current surge in demand. Recently attention has also focused on the environmental consequences of energy conversion, namely atmospheric pollution and global warming. In parallel with this, increasing emphasis is being placed on human comfort within buildings and on the quality of the indoor environment.

Energy use strategies to meet these needs include the use of less polluting energy sources, control of harmful emissions and increased energy efficiency. Energy conservation is a most effective measure in this regard, in that it simultaneously leads to reductions in energy use and environmental improvements, particularly the reduction of carbon dioxide emissions that contribute to global warming.

In Europe and the United States over 50% of all energy use can be associated with buildings [1, 2] and a considerable portion of this is consumed to moderate internal environmental conditions - more than 60% in the United Kingdom, for example [1]. Rousseau and Mathews [3] have estimated that more than 10% of *all* energy consumed in the world is expended in building heating, ventilating and air-conditioning (HVAC) systems, and it is stated that this sector is growing rapidly.

Energy awareness in the design of new buildings and retrofitting of existing buildings can, therefore, lead to substantial and widely felt benefits. It is commonly agreed that energy savings of between 30% and 70%, relative to the 1973 figures, are achievable through use of improved design methods and new technologies [1]. Another source suggests that better design of new buildings could result in a 50% reduction in energy consumption while a 25% saving could be possible through retrofit of existing buildings [2].

A recent study by Cole and Kernan [4] confirms the dominance of building operating energy over embodied energy for commercial buildings. HVAC and lighting accounted for approximately 85% of the total life-cycle energy use of the buildings considered. They conclude that strategies for reducing the life-cycle energy use should clearly progress first by introducing those design considerations which significantly reduce building operating energy.

Building designers have always strived to minimize project cost while maximizing the fitness of the building for its purpose. In recent decades increasing emphasis has been placed on human comfort and especially on energy efficiency. These latter design goals are sometimes in conflict and are always difficult to optimize because of the complexity of the system in question. In order to deal with this complexity effectively, building energy simulation has received growing attention in recent years.

1.2 Building energy modelling

A building is a geometrically complicated entity composed of many different constructional elements and enclosed volumes of air containing fittings, furnishings and plant. From a thermal viewpoint these constituent elements and enclosures are characterized by thermophysical properties such as conductance, capacitance and surface heat transfer coefficients. The different parts of the building and its environmental control systems are thermally coupled (frequently, with more than one other element) and exchange energy by most heat and mass transfer mechanisms. The boundary conditions (originating in casual heat loads and the weather) are spatially non-uniform, incorporate a lot of temporal variation and, once again, involve most heat and mass transfer mechanisms. Many of the energy transfer modes are non-linear, e.g. long-wave radiation varies with the fourth power of temperature, convection and infiltration are mildly non-linear and the thermal conductivity of some insulating materials has been shown to be marginally dependent on temperature.

Examples of heat transfer occurring in buildings (and mass transfer accompanied by heat transfer) would include:

- Conduction within solid building elements.
- Natural and forced convection at building surfaces and plant heat transfer surfaces.

- Longwave radiant heat transfer between internal surfaces and from external surfaces to the sky and surrounding buildings.
- Solar (shortwave) radiation to external surfaces and, via windows, to internal surfaces. This arrives as direct, diffuse and reflected radiation and its impact may be reduced by external shading, self-shading and reflection at building surfaces.
- Casual heat gains to internal surfaces and air masses brought about by people and heat producing equipment.
- Infiltration driven by buoyancy and wind pressure.
- Air movements within the building driven by pressure and temperature differences and by mechanical plant.
- Heat injection and removal by HVAC equipment (usually automatically controlled).
- Various forms of latent heat transfer occurring at chillers, cooling coils, steam boilers, humidifiers, cooling towers, ice thermal storage plant and even at building elements impregnated with phase change materials so as to trim peak cooling loads.

The building and its HVAC plant can be modelled by a large, non-linear set of coupled differential equations. The behaviour of the solution in time simulates the thermal performance of both. The exact (general) solution cannot be found and recourse must be made to approximate methods which, despite their description, can provide a very accurate but *particular* solution. Application of these methods involves solving a related discrete problem rather than the original continuous one. Inevitably, this requires subdivision of the problem. It might be felt that finer subdivision, made possible by ever increasing computing power, would always improve solution accuracy. This cannot be assumed, however, as some discrete methods can introduce instabilities not present in the original problem. This aspect will be discussed further in Chapter 4.

Assuming stable solution methods and sufficient computing power to allow

(i) an adequate subdivision of the problem,
(ii) full treatment of non-linearities,
(iii) inclusion of all coupling between the chosen nodes and
(iv) detailed physical modelling of the various energy storage and exchange processes

building energy modelling offers the designer an emulation of reality. Temperatures and energy flows at all points of interest within the building and its environmental control systems are available at closely spaced points in time. Gross air movements between internal spaces, as well as air exchange with the exterior, are also obtainable but details of the air movement within any space are not. This latter problem belongs in the domain of computational fluid dynamics and is different from the building energy problem in terms of the amount of computation required and the character of the describing equations. The two problems can be solved in parallel, exchanging information at intervals [5]. This thesis is concerned with building energy modelling only. Virtually any question which might arise in this domain can be answered with the aid of the model including the often quoted 'what if' questions which allow building designers to optimize the design.

The principal benefits of environmental modelling, as outlined in Chartered Institution of Building Services Engineers (CIBSE) [6] and elsewhere, are:

1. Improved energy and environmental performance of buildings. Modelling allows more accurate estimation of peak thermal loads through a fuller consideration of thermal storage, infiltration and other heat transfer mechanisms. The designer can trim safety margins with confidence and the consequent reduction in cooling and heating plant sizes leads to less use of ozone-depleting refrigerants and lower emissions of green-house gases. Modelling can, simultaneously, be an effective mechanism for optimizing internal environmental conditions by providing details of air temperature, shortwave and longwave radiation in each space. These are the main components of a thermal comfort index.

2. Evaluation of innovative design concepts. Traditional load estimation methods are often not flexible enough to deal with new building/plant configurations such as chilled concrete slabs or precooling of buildings by use of night ventilation. Designers and clients require reassurance that these systems will work in their new buildings.

3. Detailed analyses for design optimization. A change in glass reflectance or absorptance reduces summer cooling load but also reduces daylight levels and the availability of passive solar gains in winter to offset heating demand. What are the optimum glass factors? How early can a building heating system be switched off without affecting comfort? Can a building's thermal mass be used to shift the peak cooling load outside of the occupied period? Modelling can provide answers to these questions.

4. Reduction of life-cycle costs. Modelling allows estimation of the annual energy consumption for any building/plant combination and so it can be used to optimize the form

and fabric of a building together with its mechanical and electrical plant, within the constraints set by the client.

5. Ability to assess HVAC system and control performance to ensure that comfort and energy costs are acceptable mid-season as well as on 'design' days.

6. Assessment of overheating risk in ventilated buildings.

1.3 Energy modelling techniques

Design tools for the estimation of thermal loads and free-running temperatures can be classified as follows (in order of increasing complexity and achievable accuracy):

(a) Steady state methods

These calculations quantify heat transmission through solid building elements by use of U-values or thermal transmittance coefficients. No account is taken of solar gains, casual gains or thermal storage within the building. The method is, therefore, best applied for heating plant sizing in continuously heated buildings where an extended cold spell with little sunshine might be expected to determine the 'design' heat loss. Dynamic response, when the building is intermittently heated or when rapidly changing thermal loads are imposed, is not well modelled.

(b) Simplified dynamic methods

These are, essentially, manual methods which take some account of building dynamics and thermal storage. The methods are often based on multiple runs of more powerful techniques and are presented in tabular or graphical form. The CIBSE admittance procedure [7], based on harmonic analysis, can be included here. An early example of this approach would be the Carrier load estimating method [8] in which analogue computer calculations using Schmidt's method were used to tabulate dynamic heat flow through walls. The intended purpose of most of these methods is the identification of peak thermal loads rather than simulation.

(c) Dynamic methods

Most energy flow paths are modelled in these methods. There is sufficient subdivision of the problem to provide a detailed thermal history of building and plant at adequate resolution. There is strict compliance with the conservation laws. The coupling between different parts of the building and between building and plant is accounted for by solving the describing equations as a system, i.e. simultaneously. The two major techniques in this category are response function methods and finite difference methods. These and some other techniques (transfer function methods, state-space models, electrical analogue methods, artificial neural networks and genetic algorithms) will now be described.

The *response function* approach has two branches, time and frequency responses, the first of which is prevalent in North America. Frequency-domain response or harmonic analysis has been adopted by the CIBSE, in simplified form, for its manual load estimation method. Recognizing that exact solutions to complex, transient, heat transfer problems are not available, response function methods operate by building up approximate solutions to real building energy problems from analytical solutions for relatively simple constructional elements and boundary conditions.

For example, when using the time-domain approach, the exact response of a building element to a unit temperature or flux pulse [called the unit response function $u(t)$] is found by use of Laplace transformation. An arbitrary, time-varying, thermal excitation $e(t)$ can be resolved into a series of such pulses and the overall thermal response to it $r(t)$ established by superimposing individual responses:

$$r(t) = \sum_{i=0}^{\infty} u(i\Delta t) e(t - i\Delta t) \tag{1.1}$$

Here $u(t)$ has been represented by a time-series of response factors, each separated by a time increment Δt. The summed response is made up of these factors, each scaled by $e(t)$ which is evaluated with an appropriate time adjustment. The present day form of the response function technique is associated with the names of Stephenson and Mitalas [9]. It can be used to estimate room loads and temperatures using unit response functions derived for multi-layered constructions.

The frequency-domain approach entails representing the thermal driving forces by Fourier series, i.e. a constant term plus a series of sine and/or cosine functions. Exact solutions are available for single sinusoidal excitations and the principle of superposition is invoked to allow the system response to be obtained by summing the individual effects of the separate harmonics. The CIBSE admittance method, based on the work of Milbank and Harrington-Lynn [7], is a simplification of this approach in which only a single frequency is considered – the 24 hour daily cycle.

Early, extended accounts of these techniques are to be found in Kimura [10] and Muncey [11]. A most detailed description and critique of the methods is available in Clarke [12].

To illustrate the relationship between all of the dynamic methods, the following example is introduced:

$$ar'' + br' + cr = e(t), \qquad r(0) = r'(0) = 0 \tag{1.2}$$

The input (driving force) is denoted by $e(t)$ and the output (response of system) by $r(t)$. Time-division of $e(t)$ for the first approach leads to the series

$$e(t) = e(0),\ e(\Delta t),\ e(2\Delta t),\ e(3\Delta t), \ldots \ldots \tag{1.3}$$

each with domain interval Δt, whereas frequency-splitting of $e(t)$ for the harmonic approach gives

$$e(t) = a_0 + \sum_{n=0}^{\infty} a_n \cos\frac{2\pi nt}{T} + \sum_{n=0}^{\infty} b_n \sin\frac{2\pi nt}{T} \tag{1.4}$$

where T is the period of $e(t)$. Second derivatives rarely arise in building energy simulation except, maybe, in the context of control. The example is intended to demonstrate mathematical connections rather than model a particular reality. The Laplace transform $L(f)$ of any function $f(t)$ will be denoted by an upper case letter and is defined by

$$F(s) = L(f) = \int_0^{\infty} e^{-st} f(t)\,dt \tag{1.5}$$

Integrating by parts, it can be shown that the Laplace transforms of successive derivatives of $f(t)$ are

$$L(f')=sL(f)-f(0)=sF \tag{1.6}$$

$$L(f'')=s^2L(f)-sf(0)=s^2F \tag{1.7}$$

Taking the Laplace transform of both sides of Equation (1.2) yields:

$$as^2R+bsR+cR=E(s) \tag{1.8}$$

$$\therefore R(s)=U(s)E(s) \tag{1.9}$$

where

$$U(s)=\frac{1}{as^2+bs+c} \tag{1.10}$$

is the so-called transfer function which depends neither on $e(t)$ nor on the initial conditions. It is a function of the system parameters (a, b and c) only and it relates output to input in (the transformed) s-space. The inverse transform of E is e. From Equation (1.9), the inverse of the transfer function U, if it can be found, is clearly the unit response u to an arbitrary input $e(t)$. (For this example, it can be found using partial fractions and a table of Laplace transforms). The inverse transform of the product UE is established by use of the convolution integral:

$$r(t)=\int_0^\infty u(t)e(t-\tau)d\tau \tag{1.11}$$

which is normally evaluated numerically, as in Equation (1.1), to give the output or system response.

In the mid-1970s, when dynamic methods began to be developed, computer power was limited and the *response function* techniques offered short computation times. This was achieved by pre-calculating unit response functions for commonly occurring constructions. Then, only a series of products and summations are required to calculate a load. This can also be viewed as a disadvantage, of course, since the user is limited to these same constructions [3] and to the fixed time-step used in the calculations performed on them [6]. There are some other disadvantages. Time-domain methods will not, without enhancement, provide intra-constructional temperatures which might be used to estimate thermal storage or quantify the risk of interstitial condensation [13]. They are most suitable for use in constant temperature spaces, e.g. air-conditioned buildings. Special techniques are necessary if the room temperature is not constant [14]. Frequency-domain methods are not very appropriate for modelling thermal excitations such as those presented by casual gains and plant operation which do not vary smoothly and therefore require many harmonics to represent them well [13]. The CIBSE admittance procedure, which uses just one frequency, is more applicable to free running buildings than to air-conditioned ones [14]. The simplified dynamic methods of both the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), based on response factors, and the CIBSE (the admittance method) are reported to underestimate the effect of thermal storage resulting in predicted peak loads that are up to 25% higher than actual loads [3, 6]. The principle of superposition, applied in both branches of the response function approach, assumes linearity in the governing differential equations. Sources of non-linearity in the building equations are outlined in Section 1.2 and many plant models are also non-linear. Unit response functions could be computed anew at each time-step but this would negate the principle advantage of these techniques, namely their speed [1].

ASHRAE has adopted the heat balance method (HBM) which is formulated using conduction *transfer functions* as its baseline procedure for cooling and heating load calculations [15]. ASHRAE's simplified radiant time series method is derived from the HBM. The example above demonstrates the close connection between the transfer function and the response function methods and, therefore, similar comments apply.

Prior to the solution of higher-order ordinary differential equations (ODE) like Equation (1.2) by a *numerical method*, it is normal to decompose them [and also partial differential equations (PDE) in the building energy context] into a set of first-order ODEs:

$$\begin{aligned} & r' = q \\ & & r(0) = q(0) = 0 \qquad (1.12) \\ & aq' + bq + cr = e(t) \end{aligned}$$

A larger linear set of this kind might be written in matrix form:

$$\mathbf{r}' = \mathbf{A}\mathbf{r} + \mathbf{B} \tag{1.13}$$

and if the right hand side were non-linear:

$$\mathbf{r}' = \mathbf{f}(t, \mathbf{r}) \tag{1.14}$$

A variety of numerical methods can be applied, the most generally used coming from the family represented by the theta method:

$$\mathbf{r}^{j+1} = \mathbf{r}^{j} + k\left\{\gamma\mathbf{f}\left(t^{j+1}, \mathbf{r}^{j+1}\right) + (1-\gamma)\mathbf{f}\left(t^{j}, \mathbf{r}^{j}\right)\right\} \tag{1.15}$$

A discrete model such as Equation (1.15) can be generated from the original continuous model by two means [16]:

1. Direct replacement of derivatives and functions by their finite difference forms.

2. Sub-division of the building and plant, and application of the conservation laws to each finite volume of material.

Finite difference techniques are considered the most general [1] and the most flexible [17]. Calculations are from first principles and not from reference to stored data or previous computations. Non-linear phenomena can be modelled. Innovative building/plant concepts can be handled. This approach generally requires more computation but the benefits obtained and the reducing cost of computer power are leading to increasing use of this means of simulation.

A *state-space* system representation has been used in building energy simulation [18-20]. State-space techniques and transfer function methods are better known in the context of controls, the former being referred to as 'modern control' methods whereas TFMs are called 'classical'. The state-space form of Equation (1.2) evolves from Equation (1.13):

$$\begin{aligned} \mathbf{r}' &= \mathbf{A}\mathbf{r} + \mathbf{C}\mathbf{e} \\ \mathbf{y} &= \mathbf{X}\mathbf{r} + \mathbf{Z}\mathbf{e} \end{aligned} \tag{1.16}$$

and is usually expressed in linear form for the full-scale problem. The excitation (input) vector **e** in **B** has been written explicitly and the output vector **y** has been expressed as a function of the state vector **r** and of **e**. The transfer function for the system described by Equation (1.16) can be shown [21] to be

$$\mathbf{U}(s) = \mathbf{X}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{C} + \mathbf{Z} \qquad (1.17)$$

Matrix methods can be used to solve Equation (1.16) but, because non-linearity has not been addressed, the solution would be expected to be less accurate than a numerical solution of the more representative Equation (1.14).

Electrical analogue methods make use of electrical networks to represent coupled building elements and, for small problems at least, assist in visualizing the thermal interactions. Thermal resistance h/k_s and capacitance $h\rho c$ are replaced by their electrical counterparts R and C respectively and voltage is the analogue of temperature. Solution methods from the theory of electrical circuits and even from electrical transmission line theory [22] have been employed. The electrical analogy is most often used in linear form and, since it leads to the same set of equations [Equation (1.13)] as the finite volume approach [3], matrix methods can be applied but with the shortcomings mentioned in the last paragraph. Finally, successful physical analogues of this type have been constructed [23] but this approach is perhaps not the most flexible.

Artificial neural networks (ANN) are based on simple models of the brain. Several layers of neurons (nodes) exchange information through a network of synapses (connections). The output from a neuron is determined by the sum of the incoming activations, each multiplied by a synaptic weight. ANNs can be used for prediction and pattern recognition provided they are first 'trained' for the intended application using known data. This is done by adjusting the synaptic weights so as to minimize the output error for many input data sets. Although training times can be lengthy and it can be difficult to decide on the correct network architecture [24] the technique has been usefully applied to the short-term prediction of energy use in buildings, optimal control of thermal storage and load/demand management [25].

A *genetic algorithm* (GA) models evolution by natural selection. A natural population of organisms is stressed by its environment (living and inanimate) and the least fit are underrepresented in the next generation. Sexual reproduction produces replacement individuals with new combinations of genetic material to be tested and filtered again. From time to time random gene mutations occur and are retained in the gene pool if beneficial. A GA operates on a set of digital entities by quantifying their fitness in some appropriate way and discarding a fraction of the population. New individuals are 'bred' from the fittest by copying and combining some of the digital make-up of each of two 'parents'. Mutation is also modelled lest the entire breeding stock congregate around a local fitness maximum and lose diversity. This deliberate disturbance of the 'gene pool', while necessary to escape genetic dead-ends, should be infrequent so as to allow successful gene combinations to build up in the population. The GA was originally developed [26] as a general-purpose self-adapting strategy for sampling a (usually enormous) search space. The algorithm rapidly increases the number of digital strings in promising parts of the solution space. Of prime importance is the fact that the GA in essentially a parallel algorithm, since offspring can be evaluated independently. It is, therefore, naturally suited to new generation computers featuring parallel processing. GAs have now been tested in a wide variety of uses including the control of a gas pipeline system and the design of a jet engine turbine.

An interesting recent application is to the solution of ODEs [27]. Candidate solutions are represented digitally (as a set of points) and assessed on how well they satisfy the differential equation and boundary conditions. The proposed algorithm is capable of handling linear and non-linear equations, both stiff and non-stiff. However, it does not always offer the precision of advanced conventional solvers. This would not rule out its use in building energy simulation where the requirement for accuracy is not great. Of greater concern would be the number of iterations ('generations') required for convergence. The test examples given require between 50 and 200 iterations each to achieve modest accuracy. Every iteration includes an assessment which incurs a function (differential equation) evaluation for each point of each aspirant solution. By way of comparison, the numerical methods advocated in this dissertation require typically two iterations at each solution point and two or three function evaluations per iteration so GAs do not appear to be competitive in this application.

1.4 Energy modelling programs

A recent count on the number of building energy models available [28] concluded that there were over 300 in existence. Table 1.1 below lists 16 participating programs in one of the most extensive empirical validation projects ever carried out [29]. This list is reported to include ‘well respected, detailed dynamic simulation programs’ [30] which are ‘among the best in the world’ [31].

Table 1.1 International Energy Agency validation of building energy programs

Program	Solution method(s) available	Program authors/vendors/support office
APACHE	Finite difference, explicit.	IES Ltd., UK.
BLAST	Response function.	Colorado State University (CSU), USA.
CHEETAH	Response function.	CSIRO, Australia.
CLIM2000	Finite difference, implicit.	Electricite de France (EdF).
DEROB	Finite difference, implicit.	USA.
DOE	Response function.	Lawrence Berkeley Laboratory (LBL), USA.
ENERGY2	Finite difference, explicit.	Arup R&D, UK.
ESP-r	Finite difference, implicit/explicit.	Energy Simulation Research Unit (ESRU), University of Strathclyde, UK.
HTB2	Finite difference, explicit.	University of Wales College of Cardiff (UWCC), UK.
SERI-RES	Finite difference, explicit.	Ecotope, USA.
S3PAS	Response function.	Escuela Superiore Ingenieros Industriales, Sevilla, Spain.
TASE	Response function.	Tampere University of Technology, Finland.
TAS	Response function.	Environmental Design Solutions Ltd. (EDSL), UK.
TRNSYS	Finite difference, explicit.	University of Wisconsin, Madison, USA.
TSBI3	Finite difference, implicit.	Danish Building Research Institute (SBI).
WG6TC	Finite difference, implicit.	Institute di Fisica Technica, Udine, Italy.
3TC	Other.	Facet Ltd., UK.

Following a review of practicable numerical solution methods used in this and other domains in Chapter 2, the test methodology is presented in Chapter 3. Here, the method of quantifying computational efficiency is detailed and the problem is mathematically characterised. In Chapter 4, stiffness is discussed and then the main work of investigation, development and testing of numerical methods is described. First the difference equation solver used in the European reference model (ESP) is analyzed and related methods are proposed and tested. Then suitably stable methods are introduced, tested and ranked for this application.

Chapter 2:
Literature Review

The evolution of building thermal design tools has been classified into generations by Clarke [12]. The first generation is characterised by a range of simplified calculation techniques used to quantify and assess building performance at design stage. In the mid-to-late 1970s, following on the 'oil crisis' and partially driven by it, dynamic performance of buildings was increasingly stressed in order to improve the integrity and overall accuracy of modelling methods. Initially, much effort was put into response function methods (described by Clarke as second generation) because of the limited demand they make on computer power. Numerical methods, the subject of this work, were increasingly used from the mid-1980s and design tools based on these solution methods are described as third generation. The fourth generation, dating from the mid-1990s, features quality input/output and visualisation software, and a high degree of integration and interoperability. But the basic solution strategy used in most design tools remains largely unchanged [32] chiefly because the mathematics and building physics are intimately mixed and a change of solution method would affect most of the program structure.

2.1 Numerical simulation software

The number of simulation environments and programs that are used to model energy flows in buildings and HVAC systems is very large [33-36]. An extensive collection of those utilising numerical solution techniques is listed in Table 2.1. It can be seen that a good fraction of these tools make use of explicit numerical methods [Predictor-corrector methods (PC), Euler's rule (ER), Runge-Kutta method (RK), modified Euler, the explicit finite difference method]. One of the principal properties of the building energy simulation problem is 'stiffness', characterised by a wide range of thermal time scales for the building elements (Section 3.4.3; Section 4.1), and implicit methods are widely regarded as being more efficient for the solution of stiff systems [37-45].

Table 2.1 Building and HVAC system numerical simulation software

Environment or program name	Citation(s)	Numerical method(s) used	Validation method(s) used	Comments
APACHE	[46]	Hopscotch	Inter-model comparison	
BRIS	[47]	Crank-Nicolson	Empirical validation	
BSim	[48]	The implicit finite difference method	Inter-model comparison	Contains TSBI5
DEROB	[49]	Crank-Nicolson	Empirical validation	
EKS	[50, 51]	VSVO BDF, IRK, PC, ER, LMM	Not described	
ENERGY2	[71]	'Explicit finite difference'	Not described	
ESACAP	[52]	VSVO BDF	Empirical validation via CLIM2000	Used within CLIM2000 and MS1
ESP-r	[1]	Crank-Nicolson, The implicit finite difference method	Analytical verification, inter-model comparison, empirical validation	
HTB2	[53]	The explicit finite difference method	Empirical validation	
HVACSIM+	[54]	VSVO BDF	Not described	
IDA ICE	[55-57]	MOLCOL	Inter-model comparison	
MotorLab	[58]	VSVO BDF	Not described	
Smile	[59]	TR, RK	Empirical validation	
SPARK	[60]	ER, BEM, TR, BFD4, 3 no. PC methods	Inter-model comparison	
SUNCODE-PC	[61]	'Explicit finite difference'	Empirical validation	Equivalent to SERI/RES
TRNSYS	[62]	Modified Euler	Empirical validation	
WG6TC	[71]	'Implicit finite difference'	Not described	
ZOOM	[63]	Not specified	Not described	

Most of the rest of the simulation tools use well known implicit linear multi-step methods [backward Euler method (BEM), trapezoidal rule (TR) and backward differentiation formulae (BDF)]. Much recent research work in the numerical mathematics field [64] has centred on implicit Runge-Kutta (IRK) methods and the related Rosenbrock methods in an effort to improve on the performance of linear multi-step methods (LMM). Of particular interest are the diagonally implicit Runge-Kutta (DIRK) methods which inherit most of the advantages of the parent IRK family but not its highly implicit nature which makes IRK uncompetitive. In the present study, these and other recently developed families of implicit methods are assessed for use in building energy simulation.

It would be difficult to trace all the validation work carried out on the software listed in Table 2.1 because of (i) the number of programs listed, (ii) the lengthy periods some have been in existence and (iii) the number of authors/vendors/users that might have carried out the work. However, the validation methodology used in this domain is well documented. Most early developers of simulation software attempted to compare their program's predictions with measured data from a real building. A more general validation methodology began to emerge in the 1980s and papers by Judkoff *et al* [65] and Bloomfield [66] would be representative of this trend. This methodology, with some refinements and extensions, has become the accepted standard (ASHRAE, 2001) and virtually all of the major validation exercises undertaken in recent years have used it [68-72]. The International Energy Agency (IEA) on whose work the ASHRAE standard is based [73] has recently described an ongoing research project [74] aimed at further developing this same methodology. The main elements of the method are as follows:

(a) *Empirical validation* — in which program predictions are compared to measured data from a real structure.
(b) *Analytical verification* — in which output from a program is compared with an exact solution for a simple, building-related problem.
(c) *Inter-model comparison* — in which program predictions are compared with those of other programs.

The first estimates total (including measurement) error and not just numerical error, the second is too confined to be representative and the third lacks an absolute standard.

The test methodology employed in this study allows representative problems of appropriate scale and complexity to be used for testing, and solutions to these problems of arbitrary accuracy to be generated. Because the test problems are purely mathematical and do not include measurements, the only significant error present in such a test is that associated with the numerical method under scrutiny.

2.2 Monographs

In addition to those cited in Table 2.1, there exist a number of more extended works offering reviews or evaluations of numerical methods for building energy simulation. The PASSYS project [13, 77], initiated by the Commission of the European Communities as part of its Solar

Energy R&D programme, was at the time the largest passive solar research project in the world. It set out to develop reliable and affordable test procedures for passive solar (building) components (PCS) and to increase confidence in their use. Because PCSs are strongly coupled to the building, energy simulation software programs came into focus and the validation of these became a major objective of the project. A validation methodology was formulated which included as its main components those described above, and it was tested by applying it to Environmental Systems Performance (ESP-r) [1] which was selected by the Commission of the European Communities as the European reference program for simulation of the thermal performance of buildings and passive solar systems [72]. A review of the theory of finite difference methods was undertaken but it was limited to those of the family containing the Crank-Nicolson method (the only method implemented in ESP-r at the time) which also includes the explicit and the implicit methods. Analytical conduction tests were carried out on the Crank-Nicolson scheme and while the results were generally excellent, slowly damped unrealistic fluctuations in the solution were observed for certain combinations of space and time step – leading to a recommendation that further work was required on this aspect.

The dissertation of Nakhi [78] was concerned with the development of new simulation schemes for adaptive construction modelling. One part of the work investigated the possibility of improving conduction modelling within building energy simulation packages by varying the number of nodes representing each homogeneous layer of material. This idea was tested using simple conduction problems with known solutions, i.e. analytical verification. Since the study was carried out within the ESP-r environment the only finite difference methods assessed (with different node separations and for various time steps) were those of the theta method family once more. Here again, persistent unrealistic solution fluctuations were sometimes observed with Crank-Nicolson, and more stable members of the family, such as the implicit method, were proposed as possible alternatives.

An earlier dissertation by Wright [79] examined a greater range of finite difference methods with a view to developing a model for investigating the thermal behaviour of industrial buildings. The methods assessed were (i) the family referred to earlier including the explicit, implicit, Crank-Nicolson and Douglas schemes, (ii) the three-level fully implicit scheme (3LFI) and a 'Douglas-like' variant of it and (iii) a weighted average of 3LFI over three adjacent nodes. Analytical verification was used to evaluate the methods. The three-level schemes were ruled out early and the Douglas method proved disappointing despite its small truncation error. The implicit scheme was finally chosen over the Crank-Nicolson method in order to avoid persistent oscillations for long time steps, even though the latter has a smaller temporal truncation error.

A recent text by Underwood and Yik [80], which reviews the various state-of-the-art methods for modelling energy exchange and fluid flow problems in buildings, includes a survey of numerical methods used in this domain. The explicit, implicit and Crank-Nicolson methods are the only ones reviewed for heat transfer in building envelopes. For the solution of stiff sets of ordinary differential equations that arise when modelling control systems, Rosenbrock and VSVO BDF methods are mentioned. In addition, the three general validation methods outlined above are described as being 'widely established for testing the performance of building energy calculation methods and programs'.

The number of numerical methods examined in these works is again small and the validation methodology is limited as regards testing of competing methods.

2.3 Related literature

Finally, it is worth searching the literature of related disciplines for numerical techniques used in their specific applications which may be efficient in this domain.

Table 2.2 Numerical simulation software from other domains

Environment or program name	Citation(s)	Numerical method(s) used	Domain
ACSL	[81]	VSVO BDF	General purpose
ASCEND	[82]	Various	Chemical processes
ATP-EMTP	[83]	TR	Electromagnetic transients
NEPTUNIX	[84]	VSVO BDF	General purpose
SPICE	[85]	TR, VSVO BDF	Electrical circuits

Many simulation software packages, such as ASCEND, can now be used with a variety of numerical methods because the solver is a separate module within the program. Some, such as SpeedUp [86], gPROMS [87] and IDA ICE exploit DAE (differential-algebraic equation) rather than ODE (ordinary differential equation) solvers. The two solver types are, however, closely related and the most commonly used codes for the numerical solution of DAEs are DASSL [88] and RADAU5 [89] utilising VSVO BDF and IRK respectively [90, 91].

A composite of BEM and TR called implicit improved Euler (IIE) was proposed by Hanna [92] and investigated further by Ashour [93] in the context of chemical engineering. The method is second-order accurate and more stable than the frequently used TR method.
Bank *et al* [94] developed a composite method, TR-BDF2, for the simulation of circuits and semiconductor devices which is based on TR and the second-order backward differentiation formula (BDF2). It inherits the strong stability of BDF2 without the disadvantage of being multi-step. Hosea and Shampine [95] analysed the method and proposed a related one, TRX2, equivalent to a double step of the trapezoidal rule. In the same paper they show that both methods can be viewed as DIRKs and they compare them to one of the earliest and best known DIRKs by Alexander [96]. In Carroll [97] and Carroll [98] the trapezoidal rule in TR-BDF2 is replaced by the theta method and the resulting family can be expressed in conventional or DIRK form.

A stable Rosenbrock method has been applied by Verwer *et al* [99] to photochemical dispersion problems arising in the field of air pollution modelling. The method is second-order accurate and less computationally intensive than most others of this family and may, therefore, be suited to the building energy simulation problem.

The dissertation of Sundnes [100] seeks efficient numerical solution techniques for a mathematical model describing the electrical activity of the heart. The model consists of some partial differential equations (PDE) and a large number of non-linear ODEs and bears some resemblance to the building energy model. One of the best suited methods for the application was found to be a DIRK method developed by Kvaerno [101] and first presented in an unpublished manuscript in 1992.

2.4 Objectives of the current work

The dynamics of energy flow in buildings is sometimes treated as a control application or compared with a complex electrical circuit and solution methods are drawn from well established (usually linear) theory in these areas. A more fundamental approach is to consider the building directly rather than via an analogue and formulate a detailed, dynamic energy model in the form of a set of coupled non-linear differential equations. Approximate solution methods must be used and the finite difference approach is considered to have the greatest potential because of its generality and flexibility. However, the associated computational load is still substantial even on a workstation [12]. Faster processors continue to be developed but the demand for increased computing power is never ending. In this application it would allow faster simulations at the same error tolerance or, by tightening the tolerance, greater accuracy for the same computing time. Alternatively, it could facilitate a finer sub-division of the building; for example: (i) local two- or three-dimensional modelling of thermal bridges and building junctions; (ii) further sub-division (orthogonal to the direction of heat flow) of planar surfaces such as floors; or (iii) more nodes per homogeneous material layer. Or the additional computational power might be utilized to remove some modelling simplifications such as the linearization of convection and radiation terms. One further use for it might be to include more of the building's surroundings in the model so as to portray more accurately any local distortions of the recorded weather data.

One other way of achieving these same goals is to use a solution method with greater computational efficiency; this can be thought of as the accuracy attained per unit of computational effort expended. The main objective of this work is to identify and/or develop numerical solution methods that are computationally more efficient than those commonly used in this field. This calls for (i) a discerning test methodology which is sensitive only to error due to the numerical method and (ii) a means of quantifying the computational effort required to solve a typical problem. This, in turn, requires that the building energy problem be characterised mathematically so that (a) representative test problems can be formulated and (b) a set of feasible numerical methods can be identified.

It will be seen (Section 4.3.2.2) that all of the numerical methods of the last three paragraphs of Section 2.3 possess properties which make them potentially efficient for the solution of the building energy simulation problem. In this work, they are evaluated, together with the methods commonly used in this domain, by use of complex test problems tailored to this application.

Chapter 3:

Methods of Investigation and Evaluation

A mathematical model is a set of equations (or equation) which when solved allow acceptably accurate prediction of quantities of interest. The equations are just (shorthand) statements describing the states and/or interactions of the system components. Simulation in this context corresponds to solving the equations for a set of values in the case of a steady-state system (represented by algebraic equations) or for a set of functions in the case of a dynamic system (represented by differential equations). In general, linear equations permit general (analytical, exact) solution whereas nonlinear equations do not and particular (approximate, numerical) solutions have to be estimated for each set of initial/boundary conditions.

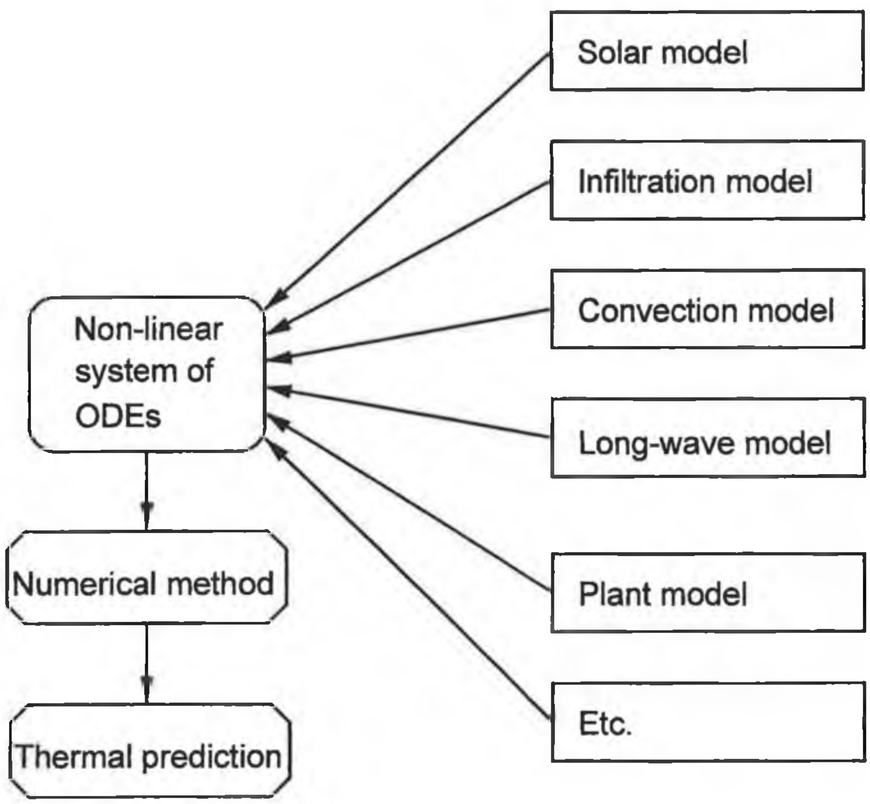


Figure 3.1 Building energy model and solution flowpath

Thermal conditions in a building are rarely steady because of the relative sizes of the thermal time constants associated with the building, its conditioning plant and the extant thermal driving forces. Coupled equations are contributed to a nonlinear set as depicted in Figure 3.1. The equations originate in building and solar physics, plant thermodnamics and control theory. Generally, they comprise conservation or 'rate' equations for energy and mass. The equations defining the building energy models used in this work are described in Appendices B and C and detailed in *Cube_acc.mcd* and *Room acc.mcd* on the attached CD ROM. Figure 3.1 highlights a point of some importance in connection with the present project: improvements in the numerical solver have a global impact on performance whereas enhancements to any of the constituent models probably have a more limited effect. The overall performance of a numerical method, as measured by its computational efficiency, dependents on the error it incurs and the computational effort expended when it is applied to problems of appropriate character. It is to these aspects we turn in the remainder of this chapter.

3.1 Classification of error

Numerous error types can be identified, all of which may contribute to an inaccurate thermal prediction. The sources of these errors range from program users to the methods and measurements they use to the machines on which they use them.

3.1.1 Input data

The dimensions of each building element include a tolerance, as do the thermophysical properties of the materials used such as density, specific heat, conductivity, emissivity and absorptivity. Weather measurements include error and interpolation of these hourly data incurs further error. Heat loads generated by people and small office equipment are statistical in nature as are the positions of window blinds, doors and other variable openings. Initial conditions, such as initial nodal temperatures, are at best estimates. The associated error can, however, be reduced without limit by including a sufficiently long pre-conditioning period. Error associated with program users has been commented on by Bloomfield [66] among others. Incorrect data may be entered; an inappropriate model may be chosen where alternatives are offered; incorrect control strategies or usage profiles may be specified.

3.1.2 *Mathematical modelling*

Modelling is an integral part of the scientific method. A hypothesis, usually in the form of a mathematical model, is put forward, tested, and improved upon or replaced if necessary. Some models, such as those describing the diffusion of heat in homogenous material or the position of the sun, are well tested and accepted. Other phenomena are inherently less easily modelled because they include a statistical or chaotic element, often in the form of fluid turbulence. This latter category includes convection at external building surfaces and solar radiation intensity through variable cloud density. However, even the first category of model may include simplification error. For example, two- and three-dimensional heat diffusion are often modelled as one-dimensional to reduce the computational load; also, assumptions about homogeneity of materials or linearity of conduction may not be valid.

3.1.3 *Numerical methods*

Stable numerical solution methods suffer from just one significant source of error. Truncation (or discretisation) error occurs because the numerical solution through a point agrees with a Taylor expansion of the exact solution through the same point for a finite number of terms only; say, terms including the p^{th} power of the step size k and less. The accuracy of such a method is said to be of order p and the error is $O(k^{p+1})$. Reducing the step size, therefore, reduces this error, and step size control to keep an estimate of the error within an appropriate tolerance is desirable, especially for stiff systems. As the calculation proceeds from point to point, the 'local errors' described above accumulate to a 'global error' which is normally $O(k^p)$ but reduces to $O(k^{p+1})$ [37, 102] for physically stable systems (Section 3.2; Section 4.1) including (generally) the building energy problem.

Compounding of error over multiple steps occurs if a numerical method is used outside of its stability region. Finally, if an interpolation method is used in conjunction with a numerical solver, it is usually chosen to be of the same order of accuracy as the numerical method. Thus, the error behaviour of the combination is no worse than that described above.

3.1.4 Computer related error

This category includes computer hardware and software errors. Roundoff errors occur because computers use a finite number of memory bits to store real numbers whereas most require an infinite number of digits for their complete specification. Single- and double-precision floating-point numbers have about 7 and 16 decimal digits of precision, respectively. The latter is built into the hardware of almost all modern microprocessors and enables more than adequate accuracy to be achieved for most practical purposes [102]. As mentioned in the last subsection, however, rounding error can be compounded and grow without limit if an unstable numerical method is used. Finally, logic errors may be programmed into an intended computer algorithm or simple coding errors may exist undetected.

3.2 Validation methodology

The accepted validation methodology for building energy simulation software, which has recently been standardised [67], has as its main elements:

(a) *Empirical validation* — in which calculated results from a program are compared to monitored data from a real building, test cell or laboratory experiment.
(b) *Analytical verification* — in which output from a program is compared with a known analytical solution.
(c) *Inter-model comparison* — in which the predictions of the target program are compared with those of other, better known, programs for the same (hypothetical) building. It is also used to make comparisons between a program and a previous version of itself – after a sub-model has been added or substituted for instance.

Empirical validation, though a necessary and appropriate application of the scientific method for whole model validation, is unsuitable for this project because it quantifies total error and not the error due to the numerical method which is sought. Inter-model comparison does not involve an absolute standard and can, therefore, be rejected for this work. Analytical verification makes use of exact solutions to simple problems usually involving heat transfer within and at the surface of homogenous layers of material [22, 75, 76]. As such, they are of the type required to test numerical methods in this application. However, being limited usually to linear problems with simple boundary conditions, they are too constrained to be representative of the problem in hand. Building energy flows are describe by a coupled set of non-linear differential equations driven by a great variety of boundary conditions some of which are not even continuous.

The test methodology employed in the present study is the one almost universally used with newly developed numerical methods in the numerical/computational mathematics literature. There, *numerical experiments* are undertaken in which the proposed method is applied to a set of test problems possessing varying degrees of stiffness, non-linearity and other properties of interest [89, 103-107]. Exact solutions are known for some of the test problems and in the case of the others highly accurate reference solutions are generated by applying a convergent numerical method with a sufficiently small time step to the problem. All useful methods, including those used to produce reference solutions in the present study, have been shown to be convergent [42, 108] and consequently the reference solution approaches the exact solution as the time step is reduced. The only significant error present in a numerical experiment, therefore, is that associated with the numerical method under test. A wide variety of problems is typically used to test a new numerical method for general use. Here we seek the computational efficiency of methods in one specific application and consequently a test problem with the mathematical characteristics of the building energy problem is formulated and variants of it are used in the evaluation process.

All of the foregoing, of course, refers to temporal convergence. Spatial convergence is also an issue in building energy simulation since some of the describing equations are functions of space as well as time (Section 4.2.1). While temporal convergence is easily confirmed (Section 4.3.3.1), spatial convergence has proved more difficult to demonstrate [109]. This is possibly because three-dimensional heat flow in buildings is usually represented by one-dimensional heat flow in sets of slabs which enclose 'lumps' of material such as air and furniture, and spatial convergence is sought in Waters [109] by reducing the nodal separation h in all of the slabs simultaneously but no subdivion of the lumps is undertaken. A second-order, central-difference approximation is almost invariably used [12, 80] to discretize the spatial derivative in the diffusion equation (Section 4.2.1). For partial differential equations, the accuracy achieved with high-order difference formulae is usually disappointing in practice [110]. Also, three nodes per homogeneous slab of masonry is considered sufficient subdivision for acceptable accuracy [12] and is often the default in simulation software. This same type and degree of spatial discretization is used here for both reference solutions and test solutions, and the reference solutions are temporally but not spatially converged. Candidate numerical methods are, consequently, tested here on differential equations in the form they normally crop up in building energy simulation.

Other test types used in the building energy simulation field include:

(d) *Sensitivity tests* — which examine the effect on output of small changes in input values.
(e) *Range tests* — which exercise the program over a wide range of input values.

The term 'sensitivity analysis' as used in the building energy simulation context describes the process of estimating bounds for the output values of a model when the input values are allowed to vary over their expected range of uncertainty [111]. If, during an empirical validation session, a particular program produces output within the estimated bounds, it is considered a success at some level. This procedure implicitly assumes that both the physical processes being modelled and the mathematical solution methods being used are stable. Since there is no experimental uncertainty in a numerical experiment, sensitivity tests are not relevant here.

If, on the other hand, extreme sensitivity to initial conditions (instability) exists, it is of interest here and the ever present rounding error is sufficient to provoke it. When it occurs it may be the result of inherent physical instability – for example, the operation of an unstable control configuration. It may also result from mathematical instability, that is an induced sensitivity due to a flawed mathematical process leading to unlimited amplification of error; examples, in this case, include ill-conditioning brought about by the use of Gaussian elimination without pivoting, and the use of an unstable numerical integrator. Numerical experiments will certainly pick up mathematical instability in the numerical method being tested because reference solutions are produced using stable methods. Systems exhibiting inherent physical instability are rarely of interest (and always difficult to simulate) but should one arise in the testing process, it will be noticed that the difference between the reference and test solutions will increase without limit.

Range tests are generally used to probe the limits of applicability of the modelling equations rather than the accuracy of the solution process. If the use of extreme input values leads to increased error or reduced stability in the numerical method under test, this will be detected in a numerical experiment because the reference solution is the product of a stable and very accurate process.

3.3 Computational efficiency

The accuracy of any convergent numerical method can be improved by reducing the step size or, in the case of an adaptive step size algorithm, by reducing the tolerance demanded. This, of course, requires additional computer effort. Accuracy, therefore, should not be the sole criterion on which numerical methods are compared. Computational efficiency (CE) may be defined as the accuracy achieved per unit computational effort expended [112]. It will be quantified here by use of the formula

$$\mathrm{CE} = \frac{1}{|\hat{\delta}| ET} \tag{3.1}$$

in which $|\hat{\delta}|$ is the maximum absolute temperature difference between the reference solution and the test solution and ET is the execution time for the test run, which includes processor operations and memory (RAM) read/write operations but not input/output operations. The maximum error $|\hat{\delta}|$ incurred during a test run is straightforwardly found. Execution time depends on the building size, the complexity of the building energy model and the type of computer processor, as well as the numerical solver in use. Since this project is solely concerned with the relative efficiencies of numerical methods, they are compared for a typical building size using a representative building energy model and performance figures (timings) for a mainstream workstation.

A typical, medium-sized building of 7500 m^2 floor area is described in Appendix D and it is shown that it can be modelled by approximately 4000 equations. Buildings of roughly this size are likely to form the majority of those requiring simulation. Since large buildings are a lot less numerous and small buildings compute very quickly anyway, a mid-range building can be justified for this work. The significant elements of computational work required to solve the mathematical model representing this building are examined in Appendix E and execution times for a single application of each on a workstation are estimated. LU decomposition (factorisation) of the system matrix is found to be the dominant computational task, even when the frequencies of the various linear algebra operations are factored in. The other tasks examined are forward/back substitution, matrix evaluation and derivative function evaluation; each requiring about two orders of magnitude less computing time than factorisation.

The detailed test problem described in Appendix C is a representative building energy model for one space. It includes most of the heat flow paths and heat transfer modes, treated at a level similar to that found in detailed simulation programs. During a typical test run with this problem the frequency of each of the significant computational tasks is recorded. They are finally multiplied by their respective execution times and accumulated to give the expected computational effort for the 7500 m^2 building. Simulation software written by others is generally unsuitable for the task undertaken here. The system Jacobian matrix (Section 3.4.3) is normally not available and would be difficult to assemble. In addition, the numerical solver is usually inextricably mixed with the rest of the program and, consequently, difficult to replace at will [32]. The chosen test problem has the required mathematical character (Section 3.4) and is, therefore, representative for the task in hand.

Performance data for a Hewlett-Packard RX2600 workstation with an Intel Itanium 2 processor are used. The Itanium 2 is representative of a new generation of 64-bit processors offering fast linear algebra. An important contributor to this performance, and one shared by many of these processors [113], is the FMAC unit which offers a hard-wired 'fused multiply-accumulate' operation so common in matrix/vector processing, e.g. in dot product and matrix multiplication. The HP RX2600 outperforms most machines in its class for a range of relevant benchmarks [114]. The processor is a joint initiative of Intel and Hewlett-Packard, two of the most respected names in the industry. Most of the major computer manufacturers have plans to use the Itanium processor in some of their high performance products [115]. Performance data for the Itanium 2 rather than the original Itanium (1) processor are used because a better match between memory bandwidth and processor floating point performance is reported for the former [116, 117]. The limitations of this project required the work to be carried out on a personal computer (PC) rather than a workstation; hence, estimated rather than measured execution times are used for the various linear algebra operations. However, the relative sizes of the estimates are consistent with expectations based on the scaling properties of the individual operations (Appendix E).

3.4 Characterisation of problem

The building energy problem is now characterized mathematically so that suitable implicit solvers can be selected for comparison and appropriate test problems constructed to evaluate them.

3.4.1 Equation types

The equation set comprising a building energy model is composed mostly of ordinary differential equations (ODE) representing the dynamic thermal behaviour of small thermal masses, e.g. room air masses, glass sheets. Continuous material such as masonry is described by the heat diffusion equation. To facilitate solution, this is semi-discretised (spatially) into ODEs which contribute to the set. Bulk air movements between internal spaces, as well as air exchange with the exterior, are usually modelled by algebraic equations (AE), as are the thermal interactions between heating/cooling plant and the building because the time constants for these processes are small. The computational burden associated with solution of these algebraic equations is small because there are relatively few of them [12, 118]. The equations requiring solution are, therefore, collectively DAEs (differential-algebraic equations) rather than ODEs, but the solution processes used for both are closely related (Section 2.3). The solution method used here, and in the building energy domain generally, is to apply a numerical integrator to the ODEs. This requires values for dependent variables in the AEs at each time step as the two equation sets are coupled. An iterative solution method is used to provide these since the AEs are generally nonlinear. Appropriate solver types are, therefore, applied in turn to the ODEs and the AEs, swapping dependent variables as required [40]. Given that the dominant task in numerical integration is the solution of a set of AEs emerging from the discretisation of ODEs, a much larger set in this case than that describing bulk air flow and plant, the computational work associated with solving the latter is relatively small.

3.4.2 Required accuracy

A relative error between 10^{-1} and 10^{-6} is frequently requested when testing numerical methods that include automatic interval adjustment [119]. An error of 0.5 K may be considered adequate for most building energy simulation work. However, since a number of error types contribute to a composite error (Section 3.1), a tolerance of 0.1 K, or 10^{-3} relative to a typical range of solution values of 100 K, is demanded of the numerical solvers under scrutiny. Hence this is a low to intermediate accuracy problem and the solution is most economically obtained using low to intermediate order numerical methods [45].

3.4.3 Spectrum

If a building energy model is represented by the vector equation $\mathbf{T}' = \mathbf{f}(t, \mathbf{T})$ and λ_i are the eigenvalues of **J**, the Jacobian matrix of $\mathbf{f}(t, \mathbf{T})$, then the set λ_i is called the spectrum of $\mathbf{J}$ and it has a large bearing on the character of the problem. The building energy problem is generally over-damped implying negative real eigenvalues. Complex eigenvalues, when they occur, can usually be traced to the plant control system and manifest themselves as oscillating temperatures or energy flows. The spectrum for the building ODE system contains a great range of values resulting from the application of the method of lines (spatial semi-discretisation) to plane slabs such as walls [42, 102], and the widely varying thermal response times of the different component parts of the building. Consequently, the ODE set is 'stiff' (Section 2.1). The extent of this property is usually measured by the stiffness ratio

$$S = \frac{Max_i \left|\mathrm{Re}(\lambda_i)\right|}{Min_i \left|\mathrm{Re}(\lambda_i)\right|} \tag{3.2}$$

Systems may be considered marginally stiff if the stiffness ratio is $O(10)$, while ratios of up to $O(10^6)$ are not uncommon.

The detailed test problem described in Appendix C is used to gauge the expected range of values of S. Descriptions of slow thermal response (heavyweight) buildings and fast response (lightweight) buildings due to CIBSE [120] are used to produce extreme versions of the test problem and many other variants between these limits. The stiffness ratio for each is determined by computing eigenvalues and can be found in Table 4.9. Approximate values of S for the various building weights are as follows: heavyweight, 6,500; mediumweight, 11,500; lightweight, 2.1×10^7. Two very responsive elements included in the lightweight building specification are chiefly responsible for its exceptional stiffness ratio; an air-gap in each partition wall and a thin aluminium facing on the external curtain wall. If the air mass, originally treated as just another thermal mass for convenience, is replaced by a more detailed convection/radiation model or by an equivalent thermal resistance [120] and if the metal facing is simply ignored because of its low thermal mass and its inability to support a temperature gradient, the stiffness ratio for the lightweight building reduces to 3,400. As a check, highly accurate reference solutions for the lightweight building, both with and without these changes, are compared using *Load & compare; 2 RUNS; light wt & light[2] wt.mcd* on the attached CD ROM. Air temperatures differ by 0.05 K and internal surfaces by 0.08 K or less. The only significant divergence between the two solutions occurs near or at the outside surface where the temperature difference is 1.2 K. If deviations at this location are not considered of importance, then the lower stiffness ratio can be used for the lightweight building. Finally, in order to confirm the stiffness ratios established by computing eigenvalues for the test problem, thermal time constants $\left(\tau = d^2/\alpha\right)$ are calculated for a representative range of construction materials [120] and for the expected range of thicknesses of each. Since $|\lambda| = 1/\tau$ the scope of S can be estimated. Identical results cannot be expected, however, since τ is a property of an individual building element whereas λ is a system property. The ratio of extreme time constants led to an expected stiffness ratio of $O\left(10^4\right)$ for the building – consistent with the above.

Accordingly, the building energy system can be considered moderately stiff and test problems with stiffness ratios of $O\left(10^4\right)$ and lower are used here to evaluate numerical solvers. In addition, the performance of each method when applied to the highly stiff lightweight building is reported but not included in the calculation of computational efficiency. Implicit methods are widely regarded as being more efficient for the solution of stiff systems (Section 2.1). However, explicit methods, used with small time increments, may be competitive when low accuracy is adequate and the stiffness ratio is not too large. Only implicit methods have been examined here.

3.4.4 Non-linearity

A set of differential equations used elsewhere to test ODE solvers includes terms up to the twelfth power in the dependent variable. The building energy model may, therefore, be described as moderately non-linear due to the presence of long-wave radiation terms containing the fourth power of temperature. Other mildly non-linear terms appear due to convection and infiltration. Also, the thermal conductivity of some insulating materials has been shown to be marginally dependent on temperature. The problem is often linearized in order to simplify it. Here, the original form of the problem is retained (when testing numerical methods for ODEs) and the search for efficiencies is considered more appropriate to the linear algebra stage of the solution which is discussed next.

3.4.5 *Dimension*

A single zone requires 50–250 nodes to represent it and a building may contain hundreds of zones. The dimension of this problem is obviously large though not as large as that encountered in the solution of partial differential equations, for example in the field of computational fluid dynamics. In the case of implicit methods, large dimension leads to an equally large set of difference equations which generally require iterative solution at each time step. Simple fixed-point iteration, if applied to this set, fails to converge unless the time increment is restricted to values comparable with explicit methods (Section 4.2.2). The Newton–Raphson method, or some variant of it, is almost always used. The most computationally expensive step in the process is the solution of linear systems involving $\mathbf{A}$, the Newton iteration matrix which is a simple function of **J** (Appendix E). Saved triangular (LU) factors of $\mathbf{A}$ are reused within the iteration loop and often for a number of consecutive time steps. When factorizing $\mathbf{A}$, advantage is taken of sparsity or any regular structure that might exist.

Dimension obviously affects the amount of computation required but not the choice of ODE solver because each of the numerical methods examined presents just one matrix for processing at each step. The same modified Newton–Raphson process is used with each of the methods.

3.4.6 *Matrix properties*

Matrices arising in this application possess few special properties, such as being symmetric or diagonally dominant [32]. Heat exchanged between two objects, for example, would lead to symmetric matrix entries if their thermal masses were the same, but this is rarely the case. Programs in the folder *Diagonal dominance* on the attached CD ROM investigate three models for diagonal dominance, (i) the detailed test problem (*Room*), (ii) the simple test problem (*Cube*) and (iii) an even simpler room model (*Dtm*) originally designed to examine control modes. The system matrices for various versions of the third model do not possess this property so diagonal dominance cannot be assumed. System matrices for the building energy problem are certainly sparse; an estimate of 5.62 for the ratio of non-zero elements to matrix order is used here (Appendix E). They also feature diagonal lines and bands expected of a three-dimensional problem though these are not as regular as might result from the discretisation of a solid. Finally, the system matrix presented at each time step is ill-conditioned as a consequence of the stiffness of the equation set [42].

3.4.7 Discontinuities

Driving forces in engineering problems are rarely entirely continuous or smooth. The building energy problem usually includes discontinuities (step changes) and discontinuous derivatives ('knee' events) in the thermal driving terms. Examples of step events are the daily switching on and off of internal lighting and plant and the operation of simple on/off control of thermal plant. Knee events can result from (i) interpolation of weather data, (ii) sign reversal in a convection term, (iii) a ceiling on the output of a proportionally controlled terminal unit and many other causes. All of those explicitly mentioned above are included in the test problems except on/off control of thermal plant – the terminal unit is controlled proportionally.

Chapter 4:

Numerical Experiments

Numerical methods, when applied to the ordinary differential equations (ODE) modelling energy flows in buildings, give rise to a system of non-linear algebraic equations (AE) known as difference equations. In Section 4.2, a frequently used direct solution method for AEs involving linearization is analysed and a related method proposed. These and one other connected method are compared using numerical experiments. Then, in Section 4.3, numerical methods for ODEs are assessed. A known instability in a commonly used method is described and alternative processes with suitable stability properties are identified. Numerical tests are used to compare a range of implicit solvers on the basis of accuracy and computational effort. But first, the central concept of stiffness and its implications are outlined in Section 4.1.

4.1 Stiffness

One of the primary influences on the selection of a numerical method for use in this domain is the extent of stiffness of the building energy model. Not only can stiffness have a large bearing on the relative efficiencies of numerical solvers, but it can effectively rule out whole families of well-known methods when it exists in significant degree. The concept has already been introduced in Chapters 2 and 3 but it will be discussed further here considering its pivotal role.

The first use of the term 'stiff' is in a paper by Curtiss and Hirschfelder [121] on chemical kinetics. It describes a differential equation as stiff if the backward Euler method (BEM) performs much more efficiently in solving it than Euler's rule (ER). A generalisation of this definition, using the relative performances of implicit and explicit methods as indicators of stiffness, is still used. The paper also introduces the backward differentiation formulae (BDF) in the context of stiff equations. At first, stiff systems of equations were thought to be unusual but it was soon realised that they are pervasive. This is reasonable in that they are associated with physical systems that exhibit a great range of time scales or, more generally, characteristic values (eigenvalues) – a much more likely event than uniform eigenvalues if the factors responsible for their magnitudes are not strongly correlated. Stiffness is not uncommon in practical problems arising in such fields as chemical kinetics, nuclear physics, process control, electronics and mathematical biology.

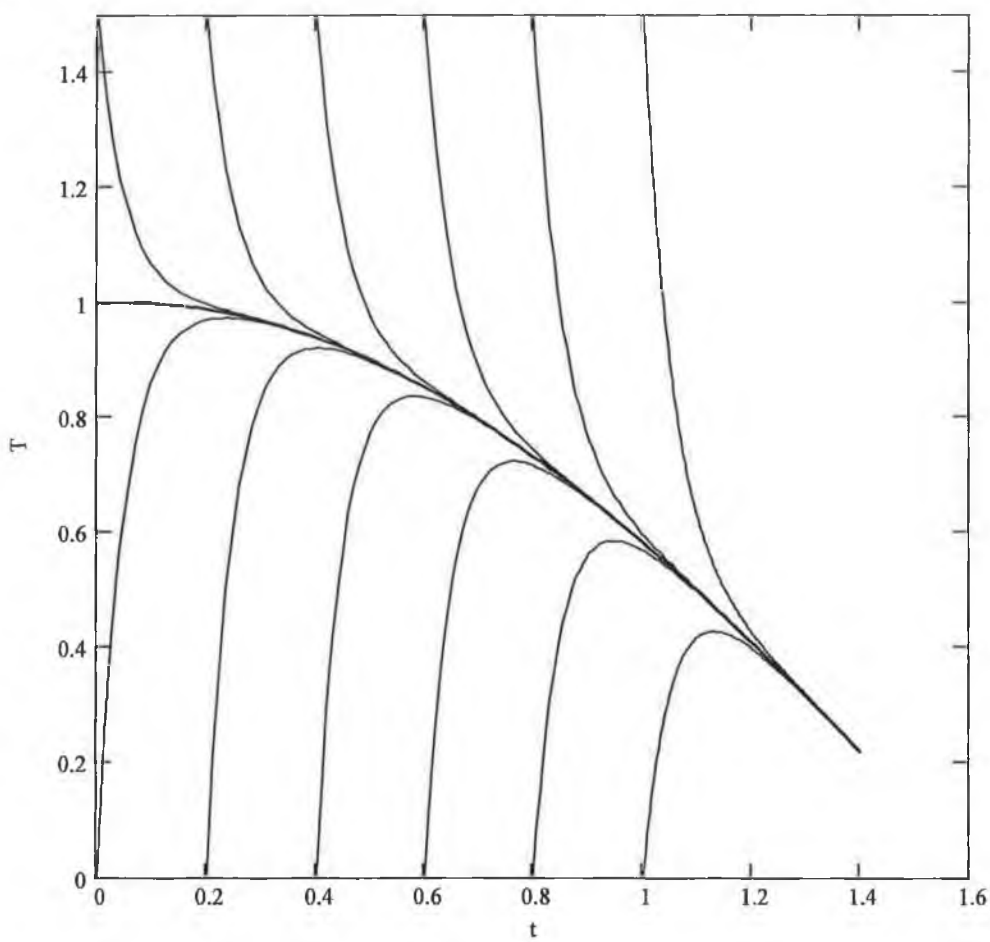


Figure 4.1 Solution space for a stiff equation, $T' = -20(T - \cos t)$

An ordinary differential equation (ODE) $T' = f(t, T)$ is stiff if $\partial f / \partial T$ is negative and

$$(b - a)\left|\frac{\partial f}{\partial T}\right| >> 1 \tag{4.1}$$

where $(b-a)$ is the integration interval. A large, negative $\partial f/\partial T$ leads to a transient solution component which converges rapidly onto the 'steady' component sought (Figure 4.1). More precisely, if the time scale $\tau = 1/|(\partial f/\partial T)|$ is small relative to the interval of interest $(b-a)$, the general solution behaviour depicted in Figure 4.1 results. We will see shortly that time steps for explicit numerical methods have to be of the order of τ for stability, resulting in very many steps and excessive computational effort when they are applied to stiff equations.

A system of ODEs $\mathbf{T}' = \mathbf{f}(t, \mathbf{T})$ is stiff if the eigenvalues λ_i of $\mathbf{J} = \partial\mathbf{f}/\partial\mathbf{T}$, the Jacobian matrix of $\mathbf{f}(t, \mathbf{T})$, satisfy the following inequalities:

$$\mathrm{Re}(\lambda_i) < 0 \tag{4.2}$$

$$Max_i|\mathrm{Re}(\lambda_i)| >> Min_i|\mathrm{Re}(\lambda_i)| \tag{4.3}$$

The extent of this property is given by the stiffness ratio

$$S = \frac{Max_i|\mathrm{Re}(\lambda_i)|}{Min_i|\mathrm{Re}(\lambda_i)|} \tag{4.4}$$

Once again, explicit numerical methods are required to take time steps approximating the fastest transient $1/Max_i|\mathrm{Re}(\lambda_i)|$ for 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 if the system is stiff.

To examine the stability of a numerical method for ODEs, the method is applied to the scalar test equation

$$T' = \lambda T \tag{4.5}$$

to get

$$T^{j+1} = r(w)T^j \tag{4.6}$$

where r is a function of $w = k\lambda$, k is the time step and λ represents $\partial f / \partial T$ if a single ODE is being analysed and it represents a typical eigenvalue in the case of a system. If an error, ε^j, exists at the j^{th} time level it will be processed through Equation 4.6 to give

$$T^{j+1} + \varepsilon^{j+1} = r(w)(T^j + \varepsilon^j) \tag{4.7}$$

Subtracting Equation 4.6 from Equation 4.7 gives the error propagation equation

$$\varepsilon^{j+1} = r(w)\varepsilon^j \tag{4.8}$$

in which $r(w)$ is described as the growth or amplification factor and sometimes the attenuation factor. Clearly the condition for error reduction, and therefore stability, is

$$|r(w)| < 1 \tag{4.9}$$

If a rational numerical method is stable when applied to Equation 4.5, it is usually stable also for the general non-linear differential system $\mathbf{T}' = \mathbf{f}(t, \mathbf{T})$ [42].

When Euler's rule (ER), $T^{j+1} = T^j + kf(t^j, T^j)$, is used to solve the test equation, $T' = \lambda T$, it gives

$$T^{j+1} = (1 + k\lambda)T^j \tag{4.10}$$

and so the condition for stability is

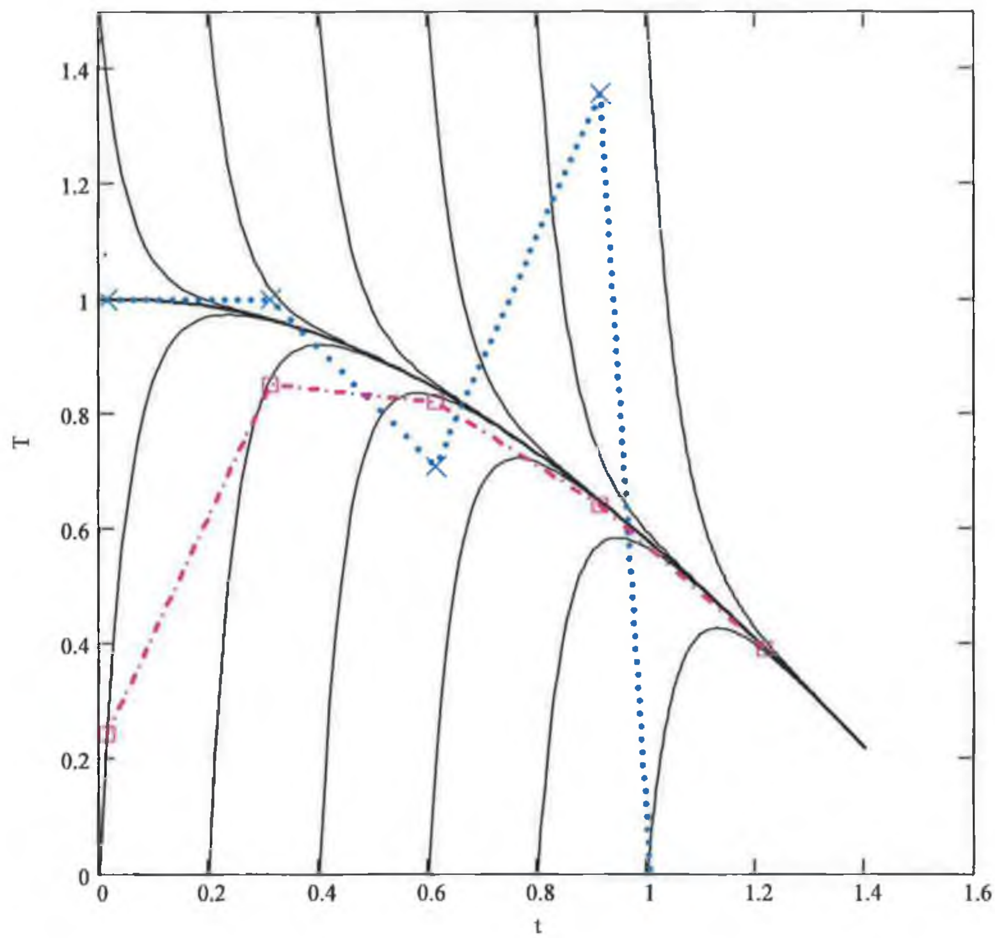
$$Max_i |1 + k\lambda_i| < 1 \tag{4.11}$$

If any of λ_i is large in magnitude, as is the case for stiff systems, k must be small to satisfy this condition. When the backward Euler method (BEM), $T^{j+1} = T^j + kf(t^{j+1}, T^{j+1})$, is applied to the test equation, it leads to

$$T^{j+1} = \frac{1}{1 - k\lambda} T^j \tag{4.12}$$

implying stability for all (negative) values of $\mathrm{Re}(w)$ and, since $\mathrm{Re}(\lambda_i) < 0$ for stiff systems, long time steps are not precluded.

Both methods are applied to a marginally stiff equation in Figure 4.2. Since $\partial f / \partial T = -20$ for the ODE, the stability criterion for ER is $|1 - 20k| < 1$ or $k < 1/10$. A time step of 0.3 is used and ER is clearly unstable. Figure 4.2, first used by Gear [45], allows graphical interpretation of the performance of the two methods. The formula for ER calls for k times the derivative (f) of the solution be added to the present solution estimate to get the next.



.....×..... Euler's rule (ER)

- - -□- - - backward Euler method (BEM)

Figure 4.2 Explicit and implicit solutions for $T' = -20(T - \cos t)$

That is, it moves from the present point to the next along a tangent drawn at the current solution estimate. But the slope of this tangent is very different from the slope of the required solution because of the stiffness of the equation and the size of the previous step. The error is thus magnified from step to step. BEM, on the other hand, moves from the present point to the next along a tangent drawn at the next solution estimate which maches very closely the slope of the solution sought. Of course, the required derivative is not explicitly available as it is for ER and so the implicit BEM equation has to be solved iteratively – a more demanding task. Graphically, this process corresponds to repeatedly drawing tangents to the solution at points on the next time line until one of them passes through the current solution point. Paradoxically then, the high stability of stiff ODEs, demonstrated by the rapidly converging solutions in Figure 4.1, leads to instability in the very popular explicit numerical methods.

4.2 An Improved Direct Solution Method

4.2.1 Model formulation and discretization

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

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{4.13}$$

The finite difference approach involves replacing the differential equations with consistent difference equations which 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 4.13 into a set of ODEs by the method of lines [42], in which space is discretized but not time. A typical nodal equation would be

$$\frac{dT_i}{dt} = \frac{\alpha}{h^2}\left(T_{i-1} - 2T_i + T_{i+1}\right) \tag{4.14}$$

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 which varies in time according to an equation of the form

$$mc\frac{dT_i}{dt} = \sum \phi(t, \mathbf{T}) \tag{4.15}$$

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 $\mathbf{T}$. A complete building energy model can, therefore, be written succinctly as

$$\mathbf{T}' = \mathbf{f}(t, \mathbf{T}) \tag{4.16}$$

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 discretization, numerical methods for ODEs are applied to Equation 4.16. For instance, the first order accurate Euler method (ER) gives the difference equation

$$\mathbf{T}^{j+1} = \mathbf{T}^j + k\mathbf{f}\left(t^j, \mathbf{T}^j\right) \tag{4.17}$$

and the second order trapezoidal rule (TR) (equivalent to the Crank-Nicolson scheme for PDEs) gives

$$\mathbf{T}^{j+1} = \mathbf{T}^j + \frac{k}{2}\left\{\mathbf{f}\left(t^j, \mathbf{T}^j\right) + \mathbf{f}\left(t^{j+1}, \mathbf{T}^{j+1}\right)\right\} \tag{4.18}$$

when applied to the same equation. The stability of any numerical method applied to Equation 4.16 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 $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{T}$, the Jacobian matrix of $\mathbf{f}$. For ER, the product(s) must lie within a unit circle in the complex plane centred at (-1,0). The size of the time increment k is, consequently, limited if ER is applied to a building thermal model for which $|\lambda_i|$ is large. It is therefore computationally inefficient for stiff systems of equations, and this is the case for most explicit methods. TR, on the other hand, is described as being A-stable because its region of stability is defined by $\mathrm{Re}(k\lambda) < 0$, that is, the whole of the left half-plane. So it is stable for all values of k but, of course, accuracy as well as stability must be considered when choosing a time increment.

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

$$L = \sup\|\mathbf{J}\| \geq Max_i|\lambda_i| \geq Max_i|\mathrm{Re}(\lambda_i)| \tag{4.19}$$

When the physical entities or processes modelled by the equations have widely differing time constants [$1/|\mathrm{Re}(\lambda_i)|$] stiffness ensues. In connection with time constants, it is worth drawing attention to a useful quantity which can be extracted from $\mathbf{J}$. It is the *pre-conditioning period* of the building represented by Equation 4.16. The pre-conditioning period is the simulation time required to allow the temperatures of all nodes to converge to values which 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 [122]. In this case an estimate is provided by calculating the time taken for the slowest transient solution of Equation 4.16 [with time constant $1/Min_i|\mathrm{Re}(\lambda_i)|$] to decay to, say, one per cent of its initial value. For the mediumweight test room used here (Appendix F), the largest time constant is $1/(2.89\times10^{-6})$ implying a pre-conditioning period of 18.5 days. This is of the same order of magnitude as an estimate in Pinney and Parand [122] for a 'modern heavyweight' domestic building.

4.2.2 Solution of difference equations

The set of equations represented by 4.18 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

$$Max_i|1+k\lambda_i| < 1 \tag{4.20}$$

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 4.18 is non-linear and so an iterative solution method is indicated. The Newton–Raphson process is the most widely accepted method for stiff systems [42]. Applied to Equation 4.18 it would take the form

$$\mathbf{T}^{j+1} = \mathbf{T}^{j+1} - \left\{\mathbf{I} - \frac{k}{2}\mathbf{J}\left(t^{j+1}, \mathbf{T}^{j+1}\right)\right\}^{-1} \left[\mathbf{T}^{j+1} - \mathbf{T}^{j} - \frac{k}{2}\left\{\mathbf{f}\left(t^{j}, \mathbf{T}^{j}\right) + \mathbf{f}\left(t^{j+1}, \mathbf{T}^{j+1}\right)\right\}\right] \tag{4.21}$$

The Newton–Raphson method converges quadratically and generally it will converge for any time increment. However, a good initial estimate for $\mathbf{T}^{j+1}$ is required. A modified, linearly convergent Newton–Raphson method is almost invariably employed in which triangular (LU) factorisation of the matrix $(\mathbf{I} - k\mathbf{J}/2)$ 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 4.18 is iterated directly for $\mathbf{T}^{j+1}$. 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 one in magnitude, in the neighbourhood of the solution. Differentiating the right hand side of Equation 4.18 with respect to each of the elements of $\mathbf{T}^{j+1}$ one gets $k\mathbf{J}\left(t^{j+1}, \mathbf{T}^{j+1}\right)/2$, leading to the convergence condition

$$\frac{k}{2} Max_i \left|\lambda_i\right| < 1 \tag{4.22}$$

Assuming Equation 4.16 is stiff, this condition restricts the time increment to values similar to explicit methods; compare with condition (4.20). It is possible, however, to rearrange Equation 4.18 so that a different iteration function appears on the right hand side. It is shown in Appendix A 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 $\mathbf{f}$ in Equation 4.18 is first decomposed in the manner done in Equation A1. For instance, the $T_2^4 - T_1^4$ expression in the longwave radiation model is factorized to give $(T_2^2 + T_1^2)(T_2 + T_1)(T_2 - T_1)$ and the first two factors are included in $\mathbf{G}$. There is no contribution to $\mathbf{g}$ from this expression. The terms are then rearranged to give Equation A4, the alternative iterative method, which is renumbered and repeated here:

$$\mathbf{T}^{j+1} = \left\{\mathbf{I} - \frac{k}{2}\mathbf{G}\left(t^{j+1}, \mathbf{T}^{j+1}\right)\right\}^{-1}\left[\left\{\mathbf{I} + \frac{k}{2}\mathbf{G}\left(t^{j}, \mathbf{T}^{j}\right)\right\}\mathbf{T}^{j} + \frac{k}{2}\left\{\mathbf{g}\left(t^{j}, \mathbf{T}^{j}\right) + \mathbf{g}\left(t^{j+1}, \mathbf{T}^{j+1}\right)\right\}\right] \tag{4.23}$$

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

4.2.3 Proposed method

Non-linear systems such as Equation 4.18, when they crop up in building energy simulation [1] or more generally in conduction modelling [123], are usually linearized before being solved by matrix inversion or some equivalent direct process. Linearization methods, such as extrapolation and lagging of dependent variables by one time step, eliminate the need for iteration. Equation 4.23 is linearized in Clarke [1] to give

$$\mathbf{T}^{j+1} = \left\{\mathbf{I} - \frac{k}{2}\mathbf{G}\left(t^{j+1}, \mathbf{T}^{j}\right)\right\}^{-1}\left[\left\{\mathbf{I} + \frac{k}{2}\mathbf{G}\left(t^{j}, \mathbf{T}^{j-1}\right)\right\}\mathbf{T}^{j} + \frac{k}{2}\left\{\mathbf{g}\left(t^{j}, \mathbf{T}^{j-1}\right) + \mathbf{g}\left(t^{j+1}, \mathbf{T}^{j}\right)\right\}\right] \tag{4.24}$$

in which the dependent variables are evaluated one time step in arrears. All the terms on the right hand side of Equation 4.24 are known and so it can be solved directly for $\mathbf{T}^{j+1}$.

Linearization 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:

1. 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.
2. The convergence factor K can be estimated using Equation A5.

3. 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 $\mathbf{T}^{j+1}$, the unknown, on the right hand side of Equation 4.23 before iteration can commence. Equation 4.24 is generated by substituting for both $\mathbf{T}^{j+1}$ and $\mathbf{T}^{j}$ in Equation 4.23 even though the latter is already known. If this unnecessary substitution is eliminated, another direct method can be put forward:

$$\mathbf{T}^{j+1} = \left\{\mathbf{I} - \frac{k}{2}\mathbf{G}\left(t^{j+1}, \mathbf{T}^{j}\right)\right\}^{-1}\left[\left\{\mathbf{I} + \frac{k}{2}\mathbf{G}\left(t^{j}, \mathbf{T}^{j}\right)\right\}\mathbf{T}^{j} + \frac{k}{2}\left\{\mathbf{g}\left(t^{j}, \mathbf{T}^{j}\right) + \mathbf{g}\left(t^{j+1}, \mathbf{T}^{j}\right)\right\}\right] \tag{4.25}$$

Notice that the proposed method requires just one vector of starting values whereas Equation 4.24 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. $\mathbf{T}^{j+1}$ in Equation 4.23 is replaced by $\widehat{\mathbf{T}}^{j+1} = 2\mathbf{T}^{j} - \mathbf{T}^{j-1}$ leading to the method:

$$\mathbf{T}^{j+1} = \left\{\mathbf{I} - \frac{k}{2}\mathbf{G}\left(t^{j+1}, \widehat{\mathbf{T}}^{j+1}\right)\right\}^{-1}\left[\left\{\mathbf{I} + \frac{k}{2}\mathbf{G}\left(t^{j}, \mathbf{T}^{j}\right)\right\}\mathbf{T}^{j} + \frac{k}{2}\left\{\mathbf{g}\left(t^{j}, \mathbf{T}^{j}\right) + \mathbf{g}\left(t^{j+1}, \widehat{\mathbf{T}}^{j+1}\right)\right\}\right] \tag{4.26}$$

4.2.4 Evaluation of numerical methods

Three direct solution methods, arising from the iterative method specified by Equation 4.23, are available for assessment. They will be referred to as LL [linearization by lagging, Equation 4.24], PM [the proposed method, Equation 4.25] and LE [linearization by extrapolation, Equation 4.26]. The Newton–Raphson (NR) method was also included for comparison purposes 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].

4.2.4.1 Computational procedures

The test problem described in Appendix B was used for the assessment. Fixed time step programs for the four numerical methods being assessed were produced and applied to this problem. Each incorporated one iteration and one matrix inversion per time step and so incurred similar computational expense. 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 [124]. 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 [41, 124], 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 – see Table 4.1 which presents accuracy statistics for a test cell of 100 mm concrete construction with an active terminal unit (test 1) and an inactive one (test 2). Other details for tests one and two are to be found in Table 4.3. The four test solutions were calculated at 15 minute intervals and also at one hour intervals. The longer time increment led to large errors in the test solutions at step changes in the casual load, especially when the terminal unit was inactive (highlighted in Table 4.1). 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 four day period following the pre-conditioning period. The statistics presented in Table 4.1 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. It is defined within each of the test programs *TR+*.mcd* on the attached CD ROM.

Table 4.1 Accuracy statistics for test runs one and two

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 4.2 Material properties

	Thickness (m)	Conductivity (W/m K)	Density (kg/m)	Specific heat (J/kg K)	Thermal diffusivity (m^2/s)
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†	2500	750	0.56×10^{-6}

† not utilized

Test runs were carried out using slabs of the first four materials listed in Table 4.2, 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 one second 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 two and eight minutes were used, the shortest time constant for 0.1 m concrete construction being five minutes in the absence of the terminal unit and less than one minute with the unit active. The casual heat gain period was also moved back and then forward by one hour so as to substantially change its time of application relative to other loads. These changes in timing were examined lest fixed relative times favour 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.

4.2.4.2 Comparison of methods

The results obtained for the test runs outlined in Section 4.2.4.1 are given in Table 4.3. Once again, the largest error occurred at step changes in the casual load when the terminal unit was inactive – the even numbered test runs in Table 4.3. 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 4.4. They measure the average factor by which the difference statistic is changed when LL is replaced by one of the other methods.

Table 4.3 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 (s)	Displacement of casual heat gain (s)	Accuracy achieved for the following numerical methods							
								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^{-2}
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 4.4 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

All three methods achieve a reduction in both mean absolute difference and maximum absolute difference when compared with LL. The decrease in maximum error is greatest for PM at 27%. Mean error reduction is greatest for NR but this method requires the construction of a Jacobian matrix or, at least, an approximation to it. PM achieves a reduction in mean error of 29%. At first sight, the performance of LE is not as good as might be expected considering its initial estimate is extrapolated from two previous solution values and should, therefore, be better than the one used with the proposed method. However, extrapolation can lead to poor initial estimates where the solution is changing rapidly, for example, at the times the casual gain is switched on or off. LL performs 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.

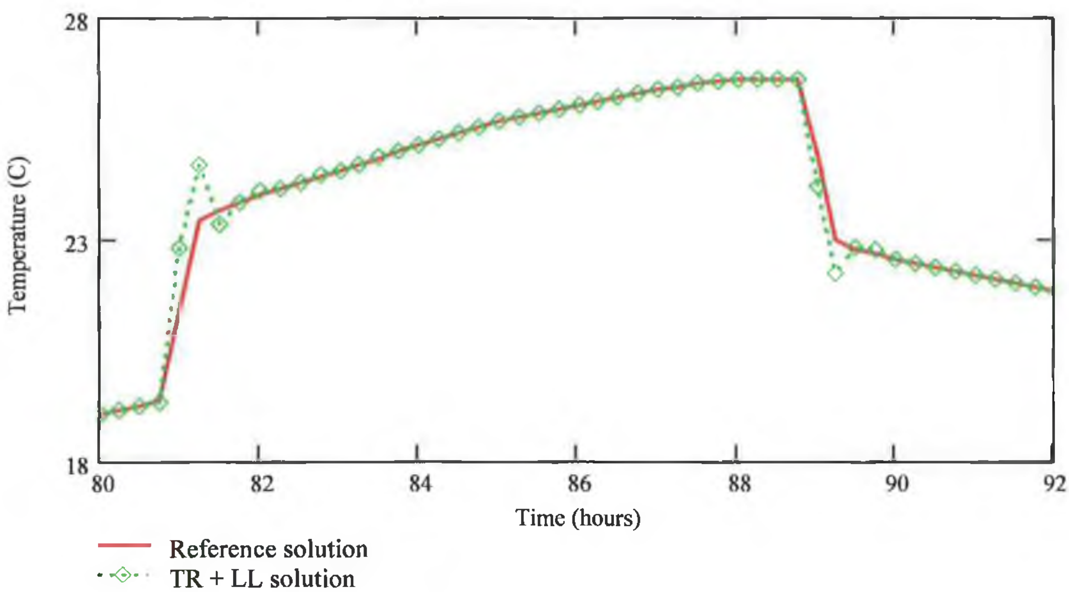


Figure 4.3 Air temperature predictions for TR + LL (Test 2, $k = 15$ min)

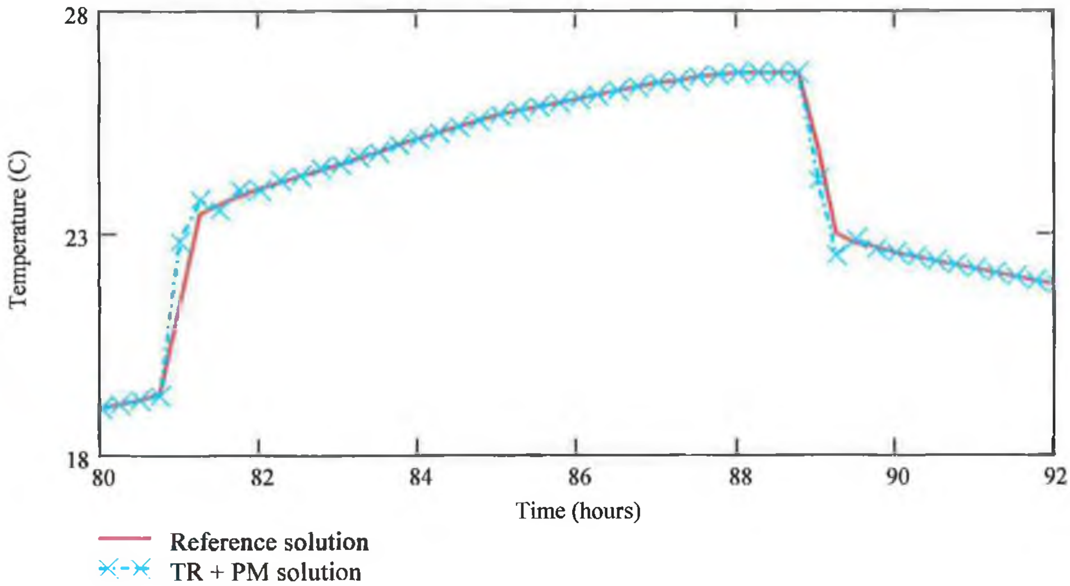


Figure 4.4 Air temperature predictions for TR + PM (Test 2, $k = 15$ min)

Figures 4.3 and 4.4 allow a visual comparison to be made between the performance of LL and that of PM over part of the interval for test run two (Table 4.3). The time step (k) is 15 minute for the test solutions and variable for the reference solutions; output for all solutions is shown at 15 minute intervals.

4.2.5 Conclusions

The use of finite difference methods to discretize the differential equations representing heat flows in buildings and elsewhere produces a system of algebraic equations which are, in general, non-linear. A commonly used linearization procedure results in a direct solution method which 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 linearization 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. 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 linearization method.

All of the tested methods can, of course, be used in conjunction with other implicit solvers for ODEs and it is to these we turn next. It is well-known that TR is just marginally stable for stiff systems and some the consequences of this for building energy simulation have been explored by Nakhi [78], Wright [79] and Waters [109]. A range of more stable methods is introduced and assessed in Section 4.3. The difference equations in each case are solved using a modified Newton iteration because it is a well-known and accepted standard and it does not require prior linearization of the problem.

4.3 Evaluation of Implicit Numerical Methods

4.3.1 Introduction

A dynamic thermal model of a building must include a means of modelling transient conduction in multi-layered building elements such as walls. The layers are most often treated as plane slabs of a homogeneous material and one dimensional heat flow is assumed. In this case Equation 4.13, the diffusion equation, together with suitable initial and boundary conditions, models the heat conduction process well. The equation and its solution are greatly simplified when presented in non-dimensional form [16]. This is done by arranging the relevant variables into suitable groups.

$$T^* = \frac{T - T_a}{T_{in} - T_a} \tag{4.27}$$

$$x^* = \frac{x}{d_{1/2}} \tag{4.28}$$

$$t^* = \frac{\alpha t}{d_{1/2}^2} = Fo \tag{4.29}$$

Equation 4.27 gives a dimensionless form of the dependent variable which must therefore lie in the range $0 \leq T^* \leq 1$. A dimensionless spatial co-ordinate is defined by dividing x by $d_{1/2}$, the half-thickness of the slab, and it satisfies $-1 \leq x^* \leq 1$. A dimensionless time is defined by Equation 4.29 and it is equivalent to the Fourier number. With these changes of variable Equation 4.13 simplifies to

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} \tag{4.30}$$

and the initial and boundary conditions become

$$T^*(x^*, 0) = 1 \tag{4.31}$$

$$\left.\frac{\partial T^*}{\partial x^*}\right|_{x^*=1} = -\left.\frac{\partial T^*}{\partial x^*}\right|_{x^*=-1} = -BiT^*(1, t^*) \tag{4.32}$$

if the slab temperature is T_{in} initially and identical convective boundary conditions exist at $x = -d_{1/2}$ and $x = d_{1/2}$. It follows that the transient temperature distribution in the slab must be of the form

$$T^* = \theta(x^*, t^*, Bi) \tag{4.33}$$

where $Bi = h_c d_{1/2}/k_s$ is the Biot number. For a given geometry, then, transient conduction is characterized by the Fourier and Biot numbers.

For most cases of interest the function θ in Equation 4.33 cannot be found exactly and recourse must be made to approximate methods involving spatial, and possibly temporal, discretization. Fundamental studies using electrical analogies have been carried out with a view to optimizing the distribution of a given number of nodes within a wall or roof, and these are summarized in Waters and Wright [125]. A number of workers considered the application of step and sinusoidal thermal excitations to the surface of a solid building element and equivalent discretized or lumped networks. It was found that the most crucial parameter governing system response was the Fourier-like dimensionless ratio $\alpha\tau/d^2$. In the case of a step change τ was the time since the step was taken and for a sinusoidal excitation τ was the inverse of its angular frequency. The smaller the value of this ratio the more difficult it was to achieve accurate modelling.

The quantity d^2/α is a characteristic time for conduction of heat through the thickness of the slab and the results above can be understood in the following way. When a thermal disturbance with a characteristic time scale, τ, is applied to the surface of a slab with a much larger conduction time scale its effects are, in the short term at least, confined to a small region near the surface. In the model, on the other hand, the disturbance is applied simultaneously to all parts of a high capacity lump and so its short term effects are diluted and unrealistic.

Waters and Wright [125] examined a family of finite-difference schemes

$$T_i^{j+1} - T_i^j = Fo_{\mathrm{fd}} \left\{ \gamma \left(T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1} \right) + (1-\gamma)\left(T_{i+1}^j - 2T_i^j + T_{i-1}^j \right) \right\} \quad (4.34)$$

which are used in many building thermal models to approximate Equation 4.13. Setting the dimensionless parameter $\gamma = 0$, $1/2$ and 1 gives the explicit, the Crank–Nicolson and the implicit schemes respectively. The mesh ratio, $Fo_{\mathrm{fd}} = \alpha k/h^2$, is a finite-difference form of the Fourier number. It was concluded that, for a given number of nodes, truncation error is minimized if nodes are distributed in a multi-layer wall in such a way that

(a) a node appears on each internal boundary between materials and
(b) the mesh ratio is everywhere the same.

Since the time step, k, is usually the same throughout, this amounts to selecting the nodal separation, h, within each layer so that the conduction time scale, h^2/α, is the same for every layer.

In the light of the above, the following strategy for the distribution of nodes in a multi-layer wall or roof would seem logical:

1. Select k to satisfy the relation

$$k = b\tau_{\min} \quad (4.35)$$

where $\tau_{\min}$ is the characteristic time scale of the most dynamic thermal excitation of interest. A small value is chosen for the constant, b, when it is required to follow the system response in detail.

2. Place a node on each internal boundary as depicted in Waters and Wright [125], and additional nodes within the layers so that the characteristic conduction time of the slice associated with each node is, as nearly as possible, the same. This time constant should be a small fraction of $\tau_{\min}$ for accuracy. The nodal separation or slice thickness, h, should therefore satisfy

$$\frac{h^2}{\alpha} = b\tau_{\min} = k \tag{4.36}$$

or

$$h = \sqrt{(\alpha k)} \tag{4.37}$$

Use of the same constant, b, leads to a corresponding subdivision of space and time. This condition can be written more simply as

$$Fo_{\mathrm{fd}} = 1 \tag{4.38}$$

This strategy merely distributes error evenly over the whole construction. To control the magnitude of the error and to avoid prolonged simulation runs, it is required to change h and k dynamically as the simulation proceeds. Changing the former essentially involves changing the number of equations in the model and is not ordinarily done. An algorithm for changing k is used in the assessment below.

4.3.2 Stability of numerical methods

Much of the earlier work, then, was concerned with local truncation error which results from replacing derivatives by finite-difference approximations. Another error type, round-off error, is inevitably introduced in computer calculations because numerical values are processed using a fixed number of significant digits. Rounding errors can normally be controlled by selective use of double-precision arithmetic unless the numerical method being used is unstable, in which case the error grows exponentially.

4.3.2.1 Commonly used methods

Crandall [126] has examined the stability and truncation error of the family of schemes represented by Equation 4.34. This work shows that large Fo_{fd} values lead to instability or oscillatory solutions unless $\gamma > 1/2$. The temporal truncation error, which is $O(k^3)$ for $\gamma = 1/2$, degrades to $O(k^2)$ for any other value of γ. The spatial truncation error is $O(h^3)$ for all γ. Hensen and Nakhi [127] have applied these results with a view to improving conduction modelling within building energy simulation packages, many of which use the Crank–Nicolson scheme $(\gamma = 1/2)$ for accuracy. Its performance under various circumstances is demonstrated in Hensen and Nakhi [127] using a test example for which an exact solution is known [128]. Homogeneous slabs with thermophysical properties and dimensions as shown in the first three rows of Table 4.2 are each represented by three nodes. One node is located centrally and represents half of the slab's thermal capacitance. Two surface nodes represent a quarter of the thermal capacitance each. The slab is initially at a temperature of 0°C, as are its surroundings. Ambient air temperature is suddenly raised to 20°C on both sides. There is no radiant heat exchange, and the convective heat transfer coefficient is assumed to be 3 W /m^2 K. The Crank–Nicolson predictions [127] for aluminium in Figure 4.5 show large temperature oscillations because of the magnitude of Fo_{fd}. Similar unrealistic temperature behaviour is predicted by the Crank–Nicolson scheme, in Figure 4.6, for a slab of insulation. In this instance a large value for $Bi_{\mathrm{fd}} = h_c h / k_s$, the finite-difference form of the Biot number, was mainly responsible for the instability [78]. The predictions for concrete ($Fo_{\mathrm{fd}} = 0.35$, $Bi_{\mathrm{fd}} = 0.16$) were quite stable with a one hour time step. Equation 4.34 with a higher degree of implicitness, up to $\gamma = 1$, is proposed [127] for use with these problematic, but commonly occurring, layers of material. The temporal accuracy of the method is, however, just first-order when $\gamma \neq 1/2$. One of the principal objectives of the present work is to identify numerical methods which are at least as accurate as the Crank–Nicolson scheme, and are stable and free of persistent oscillations in all circumstances.

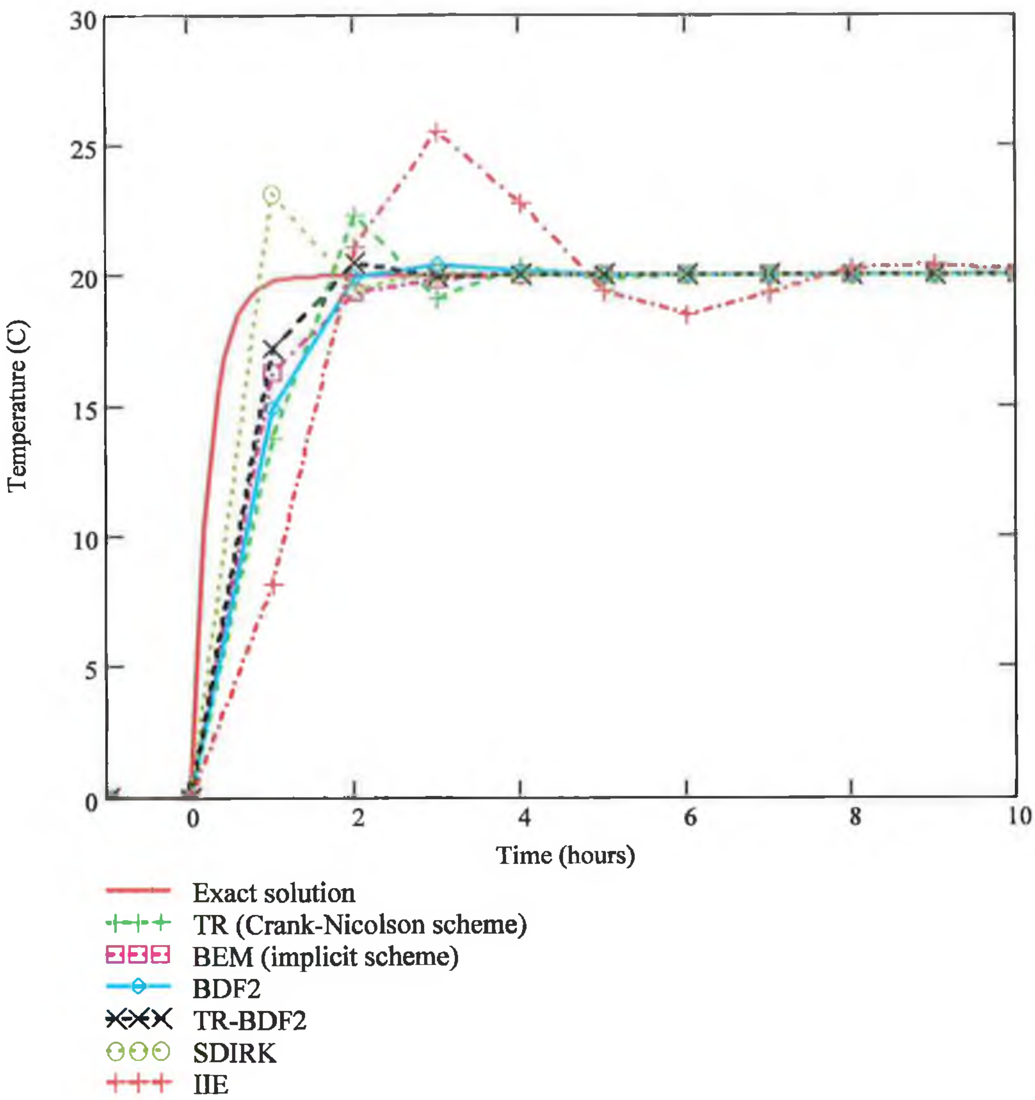


Figure 4.5 Surface temperature predictions for 2 mm aluminium using a 1 h time step

$$\left(Fo_{\mathrm{fd}} = 2.92\times10^{5}; Bi_{\mathrm{fd}} = 1.5\times10^{-5}; \max_i \left|\mathrm{Re}(k\lambda_i)\right| = 1.17\times10^{6}\right)$$

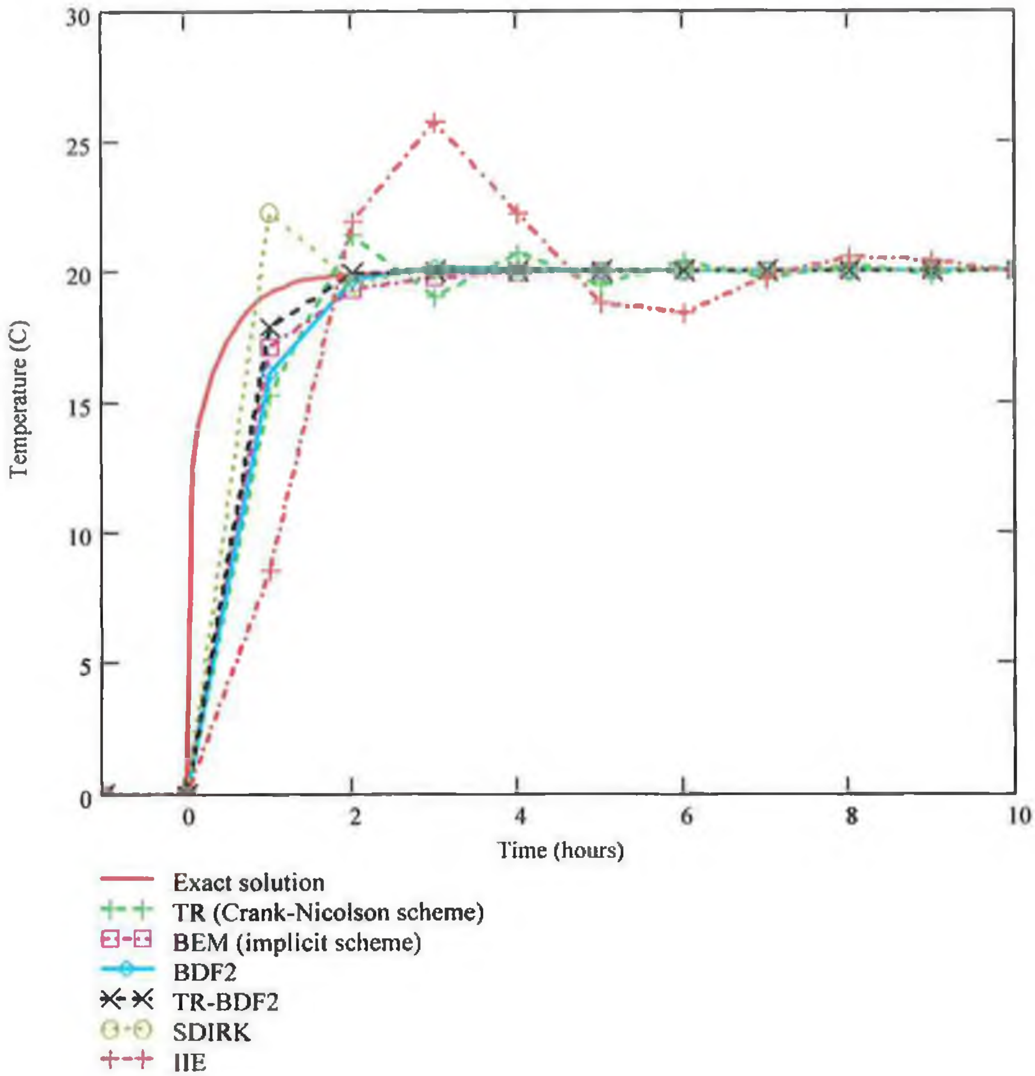


Figure 4.6 Surface temperature predictions for 100 mm insulation using a 1 h time step $\left(Fo_{fd} = 1.54; Bi_{fd} = 3.33; \max_i \left|\mathrm{Re}(k\lambda_i)\right| = 14.2\right)$

So far the discussion has centred on partial differential equations (PDE) and the accuracy and stability of finite-difference approximations to them. In Section 4.2.1 PDEs representing masonry slabs were semi-discretized and grouped with ordinary differential equations (ODE) describing the thermal behaviour of air masses and other 'lumps' of material to give a vector equation for a building energy model, Equation 4.16. If t is included among the dependent variables this equation can be written even more succinctly as

$$\mathbf{T}' = \mathbf{f}(\mathbf{T}) \tag{4.39}$$

a first-order, autonomous system of non-linear ODEs of dimension $n+1$ representing n nodes $(i = 1,2,\ldots,n)$ and time $(i = 0)$.

Numerical methods for ODEs exist which correspond to the finite difference methods previously applied to Equation 4.13. For instance, the theta method applied to Equation 4.39 gives the difference equation

$$\mathbf{T}^{j+1} = \mathbf{T}^{j} + k\{\gamma\mathbf{f}(\mathbf{T}^{j+1}) + (1-\gamma)\mathbf{f}(\mathbf{T}^{j})\} \tag{4.40}$$

which is equivalent to Equation 4.34. Setting $\gamma = 0,\ 1/2$ and 1 as before gives Euler's rule (ER), the trapezoidal rule (TR) and the backward Euler method (BEM) respectively; the ODE equivalents of the explicit, the Crank–Nicolson and the implicit schemes.

When the theta method is used to solve the test equation, $T' = \lambda T$, it gives

$$T^{j+1} = \frac{1+(1-\gamma)k\lambda}{1-\gamma k\lambda}T^{j} \tag{4.41}$$

Figures 4.7 and 4.8 show the amplification factors for the three special cases when $\gamma = 0,\ 1/2$ and 1. ER is stable in the limited interval (-2, 0). TR and BEM are stable for all (negative) values of $\mathrm{Re}(w)$ and, as such, are described as A-stable methods. Equation 4.39, when representing a building energy model , is a stiff system. A-stable methods are considered appropriate for stiff systems because large negative values of $\mathrm{Re}(\lambda)$, implied by the definition of stiffness, require small time steps, k, if ER and other methods with restricted stability intervals are to attenuate rather than magnify introduced errors. When $\mathrm{Re}(w)$ is large, in a negative sense, the amplification factor for TR approaches minus one and slowly damped oscillations result. These are apparent in Figures 4.5 and 4.6. A stronger stability property, namely L-stability, will quickly preclude these long-lived oscillations. A numerical method is L-stable if it is A-stable and, in addition, $r(w)$ approaches zero as $\mathrm{Re}(w)$ approaches minus infinity. The first-order BEM alone, of all those emerging from the theta method, possesses L-stability. All other methods assessed here are at least second-order accurate and most are L-stable.

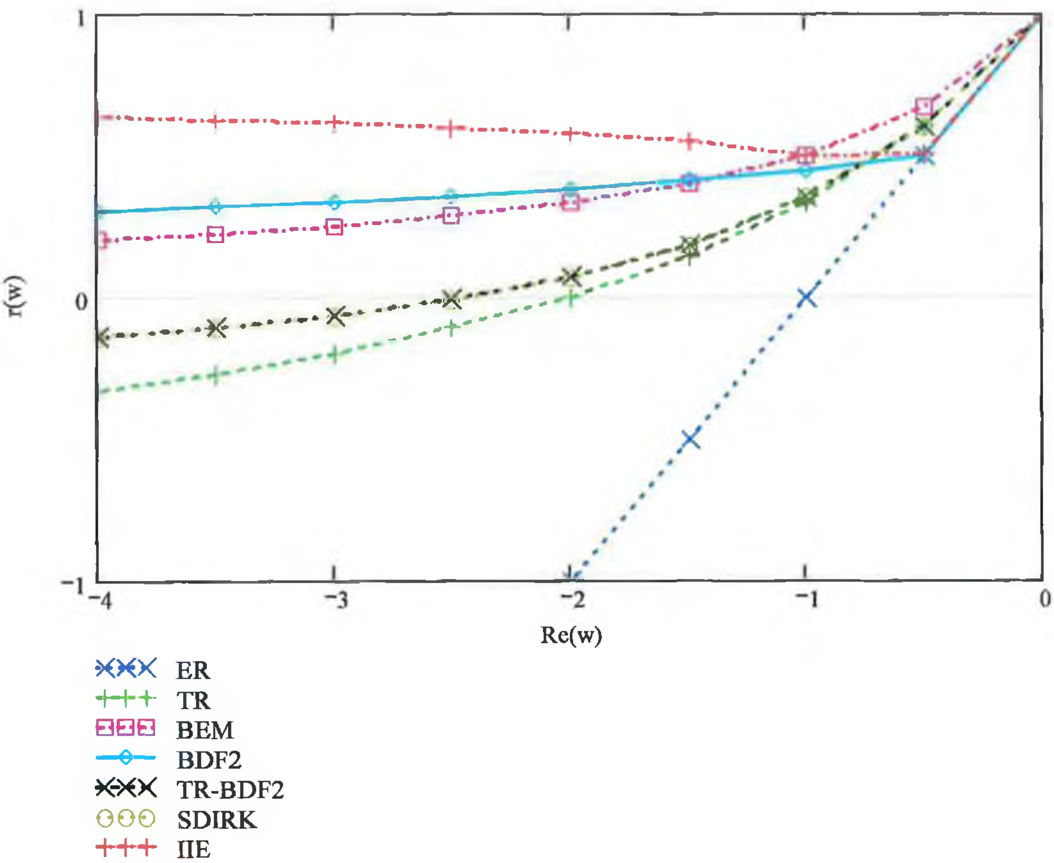


Figure 4.7 Amplification factors, $r(w)$, over a small range of (real) values for w

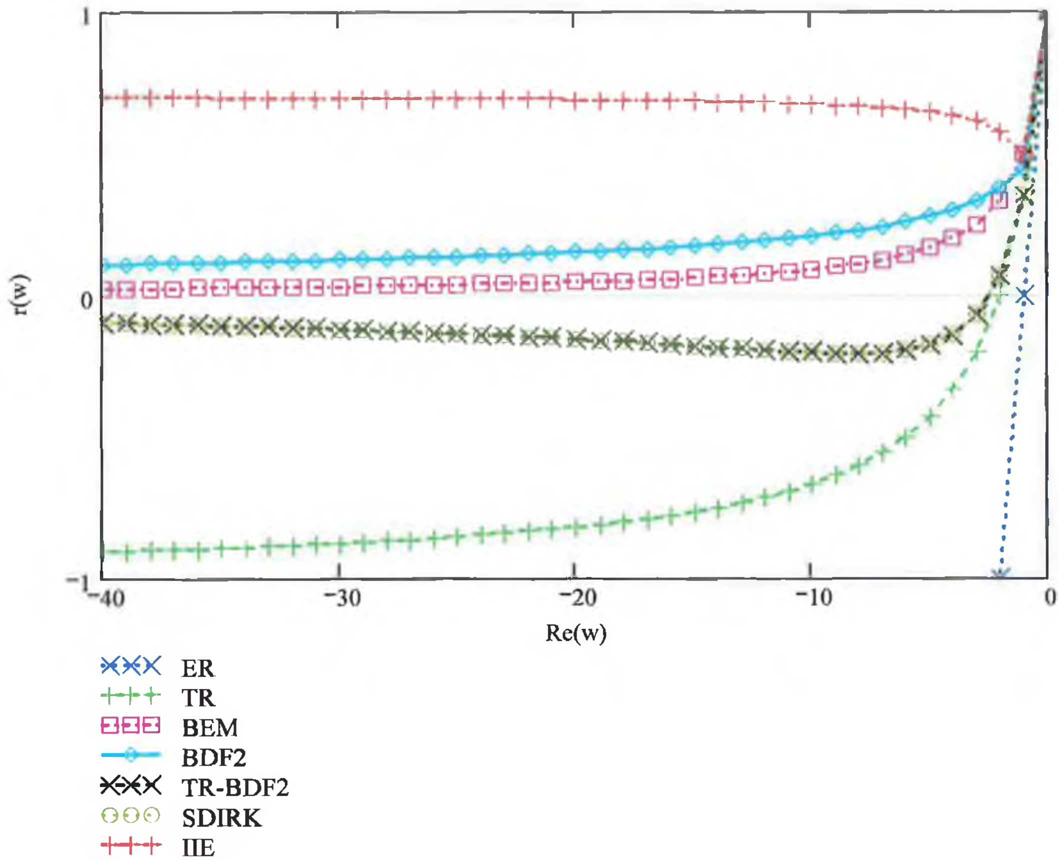


Figure 4.8 Amplification factors, $r(w)$, over a large range of (real) values for w

It is worth noting that the stiffness ratio of a system of equations, such as Equation 4.14, representing a plane slab increases as the number of nodes is increased [42, 44, 102]. As a consequence attempts to reduce spatial truncation error by reducing h can result in undesirable oscillations unless the numerical method being used is L-stable.

4.3.2.2 More stable alternative methods

The backward differentiation formulae (BDF) are among the most widely used numerical methods for stiff systems; one of the best known codes being due to Gear [45]. The second-order BDF (BDF2) applied to Equation 4.39, the general non-linear system, gives

$$3\mathbf{T}^{J+1} - 4\mathbf{T}^{J} + \mathbf{T}^{J-1} = 2k\mathbf{f}\left(\mathbf{T}^{J+1}\right) \tag{4.42}$$

and its amplification factors can be shown to be

$$\frac{2 \pm \sqrt{(1+2w)}}{3-2w} \tag{4.43}$$

The positive root is associated with the principal solution and this factor is plotted in Figures 4.7 and 4.8. BDF2 is seen to be L-stable. The BDF are not A-stable above second-order. The first-order BDF is just the BEM. All other methods examined here, except BDF2, are single-step requiring just one previous solution value to progress. A variable-step implementation of BDF2 was prepared for this evaluation.

A composite of BEM and TR called implicit improved Euler (IIE) was first proposed by Hanna [92] and later investigated further by Ashour [93].

$$\mathbf{T}_{\mathrm{BEM}}^{j+1} = \mathbf{T}_{\mathrm{BEM}}^{j} + k\mathbf{f}\left(\mathbf{T}_{\mathrm{BEM}}^{j+1}\right) \tag{4.44}$$

$$\mathbf{T}^{j+1} = \mathbf{T}^{j} + \frac{1}{2}k\left[\mathbf{T}_{\mathrm{BEM}}^{j+1} + \mathbf{T}_{\mathrm{BEM}}^{j}\right] \tag{4.45}$$

Equation 4.44 is solved implicitly at each step and this is followed by a trapezoidal improvement retaining the backward Euler derivatives. The method offers no advantage over BEM in terms of LU factorisations and derivative function evaluations. It is, however, second-order accurate and more stable than the frequently used TR. Its growth factors are

$$\frac{1}{2}\left[(1+B) \pm \sqrt{(1+B)^2 + 4B}\right] \qquad \text{where } B = \frac{w}{2(1-w)} \tag{4.46}$$

The factor associated with the principal solution is shown in Figures 4.7 and 4.8. It approaches $1/\sqrt{2}$ as w increases in magnitude and so the error damping propereties of IIE are intermediate between BEM and TR.

Bank *et al* [94] developed a composite method, TR-BDF2, for the simulation of circuits and semiconductor devices which is based on TR and BDF2. It inherits the strong stability of BDF2 without the disadvantage of being multi-step. Each step of length k consists of a fractional step of length ζk using TR

$$\mathbf{T}^{j+\zeta} = \mathbf{T}^{j} + \frac{1}{2}\zeta k\left\{\mathbf{f}\left(\mathbf{T}^{j}\right) + \mathbf{f}\left(\mathbf{T}^{j+\zeta}\right)\right\} \tag{4.47}$$

followed by a step of length k using the known values of $\mathbf{T}$ at time levels j and $j+\zeta$ in BDF2

$$\zeta(2-\zeta)\mathbf{T}^{j+1} - \mathbf{T}^{j+\zeta} + (1-\zeta)^2\mathbf{T}^{j} = \zeta(1-\zeta)k\mathbf{f}\left(\mathbf{T}^{j+1}\right) \tag{4.48}$$

The amplification factor for TR-BDF2 is

$$\frac{\left\{1+(1-\zeta)^2\right\}w + 2(2-\zeta)}{\zeta(1-\zeta)w^2 + \left(\zeta^2-2\right)w + 2(2-\zeta)} \tag{4.49}$$

Choosing $\zeta = 2-\sqrt{2}$ reduces the Newton iteration matrices for TR and BDF2 to the same form, thereby decreasing the effort required to solve the non-linear difference equations presented at each step. This value of ζ also minimizes the local truncation error and is the one exclusively used below. Hosea and Shampine [95] analysed the method and proposed a related one, TRX2, equivalent to a double step of the trapezoidal rule. In the same paper it is shown that both methods can be viewed as DIRKs. In Carroll [97] and Carroll [98] the trapezoidal rule in TR-BDF2 is replaced by the theta method and the resulting family can be expressed in conventional or DIRK form. TR-BDF2 and the method of Carroll are L-stable but TRX2 lacks this property because of its origin. All three methods are second-order accurate.

One of the earliest and best known DIRKs due to Alexander [96] takes the following form.

$$\mathbf{k}_1 = \mathbf{f}\left(\mathbf{T}^{j} + \alpha k\mathbf{k}_1\right) \tag{4.50}$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{T}^{j} + (1-\alpha)k\mathbf{k}_1 + \alpha k\mathbf{k}_2\right) \tag{4.51}$$

$$\mathbf{T}^{j+1} = \mathbf{T}^{j} + (1-\alpha)k\mathbf{k}_1 + \alpha k\mathbf{k}_2 \tag{4.52}$$

It is L-stable and second-order with $\alpha = 1 - 1/\sqrt{2}$. Its amplification factor is

$$\frac{1+(1-2\alpha)w}{(1-\alpha w)^2} \tag{4.53}$$

The method is more commonly described as an SDIRK now, in that the solution of the implicit stage equations for $\mathbf{k}_1$ and $\mathbf{k}_2$ by Newton iteration requires a single LU factorisation, the same for each stage. Methods now described as DIRKs require *s* different factorisations, *s* being the number of Runge-Kutta stages. IRKs entail the factorisation of a matrix of order *sn*, for a system of order *n*. This is often considered prohibitively expensive.

SDIRKs and other methods can suffer from *order reduction* when applied to stiff ODEs, whereby the apparent order of accuracy of the method is the stage order (often one) rather than the classical order [42]. The phenomenon is known to be problem dependent and is thus not a property of the method alone. Kvaerno [101] has constructed L-stable SDIRK methods with an explicit first stage (ESDIRKs) for which the stage order is two for large classes of stiff ODEs. A third order (classical) version will be assessed here and its structure is as follows.

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{T}^j) \tag{4.54}$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{T}^j + \gamma k(\mathbf{k}_1 + \mathbf{k}_2)) \tag{4.55}$$

$$\mathbf{k}_3 = \mathbf{f}(\mathbf{T}^j + k(\hat{b}_1\mathbf{k}_1 + \hat{b}_2\mathbf{k}_2 + \gamma\mathbf{k}_3)) \tag{4.56}$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{T}^j + k(b_1\mathbf{k}_1 + b_2\mathbf{k}_2 + b_3\mathbf{k}_3 + \gamma\mathbf{k}_4)) \tag{4.57}$$

$$\mathbf{T}^{j+1} = \mathbf{T}^j + k\sum_{i=1}^{4} b_i\mathbf{k}_i \qquad \text{with } b_4 = \gamma \tag{4.58}$$

The coefficients are next given in terms of $\gamma = 0.4358665215$.

$$\hat{b}_1 = \frac{-4\gamma^2 + 6\gamma - 1}{4\gamma} \qquad \hat{b}_2 = \frac{-2\gamma + 1}{4\gamma}$$
$$b_1 = \frac{6\gamma - 1}{12\gamma} \qquad b_2 = \frac{-1}{12\gamma(2\gamma - 1)} \qquad b_3 = \frac{-6\gamma^2 + 6\gamma - 1}{6\gamma - 3} \tag{4.59}$$

Rosenbrock methods are essentially equivalent to a single iteration of an implicit Runge-Kutta method [129]. They have been studied extensively (almost exclusively) in the engineering literature [130]. They were first developed to eliminate the need for iteration, and possible problems with convergence, when solving nonlinear problems [131]. They are direct solution methods but they require an exact jacobian implying that $\mathbf{J}$ must be evaluated and decomposed at every step. All other methods examined in this work can reuse triangular (LU) factors for many consecutive time steps (up to ten here) and the modified Newton iteration used in conjunction with them presents few convergence difficulties in practice. Since LU factorisation is the dominant computational task, Rosenbrock methods are not considered further here.

Attempts to circumvent the need for an exact jacobian began with Steihaug and Wolfbrandt [132] and since then numerous other Rosenbrock-like formulae, known as W-methods, have been put forward that allow reuse of LU factors over several time steps [99, 119, 133, 134]. Of these, only the formula of Verwer *et al* [99] evaluates $\mathbf{f}\left(\mathbf{T}^{j+1}\right)$, the derivative function at the next time point. For this reason it was the sole W-method included in the present assessment. The method of Scraton [119], for example, utilises $\mathbf{f}\left(\mathbf{T}^{j+2/3}\right)$ and, as a result, mistimes the application of discontinuities in the thermal forcing functions which occur at or near the end of the proposed step. This can lead to large error [135]. Formulae from other families of methods were found to perform poorly for the same reason; these included the implicit mid-point rule [42], the Hopscotch method [136] and the optimal second-order one-leg method [137, 138]. Consequently, only those methods that sample the thermal driving forces at the next time level are evaluated here. The details of the second-order method of Verwer *et al* [99] are as follows. It is L-stable with $\gamma = 1 + 1/\sqrt{2}$.

$$\left(\mathbf{I} - \gamma k \mathbf{J}\right)\mathbf{k}_1 = \mathbf{f}\left(\mathbf{T}^j\right) \tag{4.60}$$

$$\left(\mathbf{I} - \gamma k \mathbf{J}\right)\mathbf{k}_2 = \mathbf{f}\left(\mathbf{T}^j + k\mathbf{k}_1\right) - 2\mathbf{k}_1 \tag{4.61}$$

$$\mathbf{T}^{j+1} = \mathbf{T}^j + \frac{3}{2}k\mathbf{k}_1 + \frac{1}{2}k\mathbf{k}_2 \tag{4.62}$$

All but two of the formulae compared are second-order accurate. Methods with a low order of accuracy are likely to be the most efficient in the building energy simulation application for the following reasons.

(a) Discontinuities in $\mathbf{f}(\mathbf{T})$ and its derivatives due to step changes, knee events and the like (Section 3.4.7) mean that the higher derivatives, which are presumed to exist for higher order methods, are likely to be poorly approximated at points of disturbance in the solution.
(b) An Error-Cost plot shows that low accuracy solutions (Section 3.4.2) are most economically obtained using low order numerical methods [38, 45].
(c) The order of an A-stable linear multistep method cannot exceed two [139].
(d) High order methods may suffer order reduction (to first- or second-order) because the building energy problem is stiff.
(e) Spatial derivatives are usually approximated to second-order in this problem (Section 3.2); second-order temporal approximations would be consistent.

Nonetheless, a method of order one and another of order three are included in the assessment set.

Figures 4.5 and 4.6 show the performance of a number of these methods when applied to demanding test examples in which Fo_{fd} or Bi_{fd}, or more generally $|\mathrm{Re}(w)|$, is large. Rapid attenuation of rounding error is evident for the L-stable methods because, as is clear from Figures 4.5 and 4.6, $|r(w)|$ is small even when $|\mathrm{Re}(w)|$ is large. IIE is seen to be erratic and slow to settle.

4.3.3 Evaluation of numerical methods

The list of methods selected for assessment is as follows: BEM, TR, IIE, BDF2, TR-BDF2, TRX2 and the methods of Alexander (Alex2), Carroll (Carroll), Kvaerno (Kvaerno3) and Verwer *et al* (ROS2); the corresponding program names for the latter four are shown in brackets. Variants of these methods are also tested. As well as a conventional implementation of TR, a version that uses converged derivative functions where appropriate is included. The term 'converged' is used here to describe those $\mathbf{f}$ values produced in the process of finding $\mathbf{T}^{j+1}$ using Newton iteration as part of a stable implicit process. A conventional $\mathbf{f}$ value at the same solution point is found by evaluating it using $\mathbf{T}^{j+1}$ as argument. Because $\mathbf{f}$ for a stiff system is ill-conditioned, small errors in $\mathbf{T}^{j+1}$, such as the inevitable rounding errors, are greatly magnified. For this reason converged $\mathbf{f}$ values are normally preferred [40, 95] and are generally used here. There are some exceptions: (i) ROS2 which does not involve iteration, (ii) ESDIRK methods such as Kvaerno3 which use an explicit (conventional) first stage to mitigate the effects of order reduction and (iii) TR-BDF2, TRX2, Carroll and TR each of which includes an explicit first stage as part of its specification. Because it is so well known and so widely used, conventional TR is used as the bench mark in the present study.

A property of some methods, described as first-same-as-last (FSAL), allows one function evaluation to be saved by using the final converged $\mathbf{f}$ value in place of the explicit first stage. And, where present, it is advised that this feature be used to avoid error amplification in the first stage [95]. Kvaerno3, for example, has this property but the explicit stage is included by design to alleviate order reduction and so is left in place during testing here. It was observed during this project that use of the FSAL property in TR-BDF2, TRX2, Carroll and even TR, all of which include an explicit stage, could lead to order reduction. Since it is unclear where the balance of advantage lies, it was decided to include versions with and without the explicit first stage for several of these methods.

4.3.3.1 Computational procedures

The more detailed test problem described in Appendix C was used for the assessment. In order to control error and solve stiff differential equations efficiently, some form of interval adjustment must be used, as is evident from Table 4.1. This entails varying the time increment until local truncation error (LTE) is within a specified tolerance which was set to 0.1 K per step for this work. The principal part of the LTE for a proposed time step, k^j, is given by

$$-\frac{1}{2}\left(k^j\right)^2 T''^j \tag{4.63}$$

for BEM, and by

$$-C_{\text{lte}}\left(k^j\right)^3 T'''^j \tag{4.64}$$

for a number of the second-order methods being assessed. The error constants (C_{lte}) for these methods are given in Table 4.5. The error expressions for the other members of the assessment set are not so simple. They include derivatives different from those shown above and, consequently, are not directly comparable in this way. All of the foregoing pertains to local temporal truncation error. Local spatial truncation error is not controlled here because the space increment is constant throughout each simulation run. However, all methods are affected equally.

Table 4.5 Local truncation error constants

Numerical method	TR	BDF2	TR–BDF2	TRX2	Carroll
Rational form	$\frac{1}{12}$	$\frac{2}{9}$	$\frac{3\sqrt{2}-4}{6}$		
Decimal form	0.0833	0.2222	0.0404	0.0303*	0.0349

* An approximation [95]

Adaptive step size versions of the numerical methods were programmed, each including a routine to force a small time increment at step changes in the casual load and at on/off times for the terminal unit. These are considered to be the only predictable discontinuities. Depending on the estimated error in the solution, the proposed step, k^{J+1}, is doubled or repeatedly halved until the error is within tolerance. The limit of two placed on the factor by which the time step can be increased is an expression of conservatism when extrapolating; it helps limit expensive step failures. An initial step size of the order of the smallest time scale $\left(1/|\lambda|_{\max}\right)$ in the problem was used [129].

ROS2 does not require iteration; for all other methods a modified Newton-Raphson method was used in which the Newton iteration matrix is updated and inverted at least once every ten time steps, but not normally within the iteration loop. More frequent updating can be triggered by failure of the iteration to converge, which occurs when k is too large or when $\mathbf{J}$ needs updating [40]. Since both require a subsequent, expensive decomposition of the iteration matrix, k is halved and $\mathbf{J}$ is re-evaluated whenever the Newton iteration fails to converge. Previously [135] it was found that just one iteration per time step was generally adequate for the simple test problem provided the initial approximation was generated using Newton's divided difference interpolation formula. The use of the more detailed test problem and especially the inclusion of less robust methods in the test set has necessitated a change in strategy. An extrapolated initial estimate led to poor results for some methods so $\mathbf{T}^J$, the latest solution value, was used to seed the next iteration for all methods. A number of methods performed poorly when a fixed (low) number of iterations per time step was specified, Kvaerno3 being the least stable in this regard, so the stopping criterion described in Hairer and Wanner [89] was introduced to terminate the iteration process. The implementation of ROS2 matches as closely as possible the foregoing. The system matrix is updated and inverted at least once every ten time steps. When the estimated error is exceeds the tolerance, **J** is re-evaluated; if the problem persists, k is reduced until the step succeeds.

Any change in step size is based on an estimate of the local error. Since each adjustment in k necessitates an LU decomposition, change is undertaken conservatively. Error estimation methods include:

(a) Principal local error: Expressions for the principal local error such as 4.63 and 4.64 can be used with Newton or Hermite approximations to the derivative. Sometimes f'' is approximated in place of T''' (using Equation 4.39) or f' in place of T''; when this is done converged f values are used except in the case of the bench mark version of TR.

(b) Embedding: Each step is repeated using another method of order one greater than the numerical method of interest. The difference between the two is the principal error for the lower order method. Newton or Hermite interpolation formulae can be used to provide the higher order estimate.

(c) Milne's device: The difference between two numerical methods of the same order and with principal local error expressions of the same form can be related to the error of either.

(d) Jay's device: Sometimes the only suitable secondary method available for error estimation is one or more orders of accuracy less than the primary method. Jay [140] describes a reduced tolerance in terms of the original tolerance and the orders of the two methods. If k is selected so that the error of the lower order formula (i.e. the difference between the two methods) is approximately equal to the lower tolerance, the chosen step size is appropriate for the primary method. This device is used here with IIE, Kvaerno3 and ROS2.

(e) Step-halving: Each step of length k is repeated using two steps of length $k/2$. An error estimate can be extracted from the difference. The method is rarely used now because it doubles the number of LU decompositions required.

A prime requirement of any error estimate is that it be cheap. Step-halving fares badly in this regard. Embedding, Milne's method and Jay's device appear expensive, at first sight, but the primary and secondary numerical methods are usually closely related and generally share LU factors and even derivative function evaluations. The behaviour of error estimates (E) for large step sizes (k) has received recent attention in the context of stiff systems. If, for instance, E approaches a limit or even decreases as k increases, the error control strategy will not perform as expected. Error behaviour is analysed by applying the numerical method to the simple test equation $T' = \lambda T$ once again. The error estimate for a proposed step k^j can be put in the form $E = \phi(k^j \lambda)T^j$ for some function ϕ. According to Scraton [119] ϕ should ideally satisfy the following criteria:

$$|\phi(k\lambda)| \text{ increases monotonically with } k \tag{4.65}$$

$$|\phi(k\lambda)| \to \infty \text{ as } k \to \infty \tag{4.66}$$

The more stringent criterion proposed by Hosea and Shampine [95] requires E to be asymptotically correct for small and especially large step sizes. That is

$$\left|\frac{\phi(k\lambda)}{\exp(k\lambda) - r(k\lambda)}\right| \to 1 \text{ as } k \to 0 \text{ or } \infty \tag{4.67}$$

Here, $\exp(k^j \lambda)T^j$ is the exact solution and $r(k^j \lambda)T^j$ the numerical solution of the test equation for the step in question. Criterion (4.67) is difficult to meet in practice and the filtering technique used in Hosea and Shampine [95], and first proposed by Chua and Dew [141], to ensure its satisfaction is considered in Scraton [119] to 'generally lead to a saving in computer time, but with considerable loss of accuracy'. In this study, error estimates are chosen to satisfy criterion (4.65) and especially (4.66). Criterion (4.67) is generally not satisfied except where the above-mentioned device is included as part of an algorithm – as in the cases of TR-BDF2 and TRX2 in Hosea and Shampine [95]. Table 4.6 names the variants of the selected numerical methods and describes the error estimator used with each, as well as some other particulars of each implementation. The first version listed is generally the closest to the original implementation. An exception is TR-BDF2(a) which is implemented broadly as in Hosea and Shampine [95] where the error estimate used in the original source [94] is critisized. The coded algorithms for all the numerical methods examined here, and variants of same, are to be found in the subfolder *Methods* on the attached CD ROM.

Table 4.6 Variants of numerical methods selected for evaluation

Numerical method	Important particulars of the implementation
Alex2	Hermite extrapolation used to form error estimate (embedding).
BDF2	Principal local error used, with Newton approximation to f''.
BEM	Principal local error used, with Newton approximation to f'.
Carroll(a)	Principal local error used, with Newton approximation to f''.
Carroll(b)	FSAL version. Principal local error used, with Hermite approximation to T'''.
IIE	Jay's device used to form error estimate.
Kvaerno3	Jay's device used to form error estimate.
ROS2(a)	Jay's device used to form error estimate.
ROS2(b)	Hermite extrapolation used to form error estimate (embedding).
TR(a)	Principal local error used, with Newton approximation to f'' (employing conventional f values).
TR(b)	Principal local error used, with Newton approximation to f''.
TR(c)	FSAL version. Principal local error used, with Newton approximation to f''.
TR-BDF2(a)	FSAL version. Embedded error estimator used, modified by the filtering technique of Chua and Dew.
TR-BDF2(b)	FSAL version. Principal local error used, with Hermite approximation to T'''.
TR-BDF2(c)	Hermite extrapolation used to form error estimate (embedding).
TR-BDF2(d)	Principal local error used, with Hermite approximation to T'''.
TRX2(a)	FSAL version. Embedded error estimator used, modified by the filtering technique of Chua and Dew.
TRX2(b)	Hermite extrapolation used to form error estimate (embedding).

The work was carried out on a personal computer using a general purpose mathematical software package [124]. 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 [41, 124], 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 – see Table 4.7 which presents accuracy statistics for the medium weight test problem with an active terminal unit (test 3). Other details for tests run three are to be found in Table 4.9 and a computer file referred to therein. The reference solution was subtracted from each of the test solutions in turn at every node and every hour (on the hour) over a four day period following the pre-conditioning period. The statistics presented in Table 4.7 were extracted from the set of differences for one test run. Mean absolute difference gives an overall measure of accuracy but it was felt that maximum absolute difference should be used in the calculation of computational efficiency (Section 3.3) lest a small number of unacceptable errors be concealed by the averaging process. These might be anticipated at times of rapid change in the solution. Mean difference detects any bias towards over- or under-estimation of the solution and the cross-correlation coefficient, which is defined in *Run, save & test; METHOD.mcd* on the attached CD ROM, gives a measure of the phase relationship between the reference solution and each of the other solutions.

Table 4.7 Accuracy statistics for test run number three

Numerical method	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	3.12×10^{-6}	4.64×10^{-6}	2.04×10^{-4}	1.0000
Runge-Kutta	-3.12×10^{-6}	4.64×10^{-6}	2.04×10^{-4}	1.0000
Alex2	0.00059	0.0010	0.051	1.0000
BDF2	0.00244	0.0045	0.097	1.0000
BEM	-0.00740	0.0228	0.295	1.0000
Carroll(a)	0.00100	0.0028	0.155	0.9998
Carroll(b)	0.00022	0.0030	0.155	0.9998
IIE	0.00268	0.0093	0.422	0.9997
Kvaerno3	0.00352	0.0055	0.145	0.9998
ROS2(a)	0.00609	0.0139	0.430	0.9994
ROS2(b)	-0.00005	0.0012	0.120	0.9999
TR(a)	0.00061	0.0044	0.242	0.9995
TR(b)	0.00258	0.0057	0.242	0.9995
TR(c)	-0.00018	0.0044	0.243	0.9995
TR-BDF2(a)	0.00097	0.0038	0.170	0.9998
TR-BDF2(b)	0.00028	0.0033	0.171	0.9998
TR-BDF2(c)	-0.00022	0.0017	0.087	0.9999
TR-BDF2(d)	0.00008	0.0033	0.171	0.9998
TRX2(a)	0.00219	0.0040	0.122	0.9999
TRX2(b)	0.00018	0.0012	0.060	1.0000

Table 4.8 Measures of computational effort for test run number three

Numerical method	LU decompositions	Forward/back substitution pairs	Matrix evaluations	Derivative function evaluations
Alex2	395	2788	298	2788
BDF2	414	1054	388	1028
BEM	330	1202	280	1152
Carroll(a)	206	1248	188	1542
Carroll(b)	205	1124	194	1124
IIE	267	911	238	882
Kvaerno3	208	1576	208	1838
ROS2(a)	243	732	230	691
ROS2(b)	427	2018	358	1904
TR(a)	268	850	243	1248
TR(b)	272	863	245	1237
TR(c)	272	879	239	1252
TR-BDF2(a)	197	1398	185	1228
TR-BDF2(b)	191	1120	177	1386
TR-BDF2(c)	379	2469	312	3019
TR-BDF2(d)	199	1132	186	1402
TRX2(a)	187	1255	183	1008
TRX2(b)	394	2552	303	3099

Each of the test programs was equipped to keep a tally of the most expensive steps in the solution process. They are LU decomposition (factorisation), forward/back substitution, matrix evaluation and derivative function evaluation, and execution times for a single application of each are estimated in Appendix E. Table 4.8 lists these measures of computational effort and gives their frequencies for a single test run. The totals for each are finally multiplied by their respective execution times and accumulated to give the expected computational effort for the typical 7500 m^2 building described in Appendix D.

Ten variants of the test problem described in Appendix C were used for the assessment. Descriptions of slow thermal response (heavyweight) buildings and fast response (lightweight) buildings due to CIBSE [120] were used to produce extreme versions of the test problem and the rest lie between these limiting cases. The stiffness of the lightweight building was exceptional so minor changes were made to its specification with modelling consequences which may be considered acceptable (Section 3.4.3). This reduced its stiffness ratio to the same order of magnitude as the rest. Results for the original fast response building specification were not used in the assessment process but are reported nonetheless. The different versions of the test problem include building elements of various materials and thicknesses. The pre-conditioning period changes and so different thermal driving forces (weather data) are applied. Internal heat gains were varied, as were the set point and proportional band of the terminal unit. On/off times for both the casual heat gain and the terminal unit were moved back and forward. Discontinuities in the heat gains were expected to lead to the greatest thermal disturbance so the tests were carried out with these on/off times occurring just before some of the assessment points. The time gap (prior to assessment) was set to the shortest time scale $\left(\tau_{\min} = 1/|\lambda|_{\max}\right)$ for the particular variant of the problem. Limited experimentation indicated that numerical error peaked after a delay of this order. Values for $\tau_{\min}$ were found to be in the range one half to two minutes. 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. Further details of the variants used are to be found in a file named *Building types.mcd* on the attached CD ROM.

4.3.3.2 Comparison of methods

The performance of a numerical method should be judged not just by the accuracy achieved but also by the computational effort expended because one can usually be traded for the other. The measure of computational efficiency used here is $\mathrm{CE} = 1/\left(|\hat{\delta}|ET\right)$, where $|\hat{\delta}|$ is the maximum absolute temperature difference between the reference solution and the test solution (Table 4.7) and ET is the execution time for the test run. The results obtained for the test runs outlined in Section 4.3.3.1 are given in Table 4.9. In contrast with Table 4.3, the results achieved for the different problem variants (Table 4.9) did not generally deteriorate when the terminal unit was switched off because error is controlled in this case by interval adjustment. CE is not a smooth function because of the use of peak rather than average temperature deviations, but the number of tests undertaken allows a statistical comparison of the performances of the numerical methods. To this end the CE for the bench mark method, TR(a), was divided into the CEs of each of the other methods in turn. The geometric mean values of these ratios are presented in Table 4.10 for a short list of methods. Some poorly performing methods and variants were eliminated as testing progressed. Some others, such as TR and BEM, were retained for comparison, despite their relatively low efficiencies, because they are so widely used. CEs were calculated for the full set of ten test runs in the case of the nine short-listed methods only. The factors in Table 4.10 represent geometric mean improvements in computational efficiency over TR(a) for each of the short-listed numerical methods. Alex2 [96] is seen to be more efficient than TR(a) by a factor of 4.27.

Table 4.9 Computational efficiency for the test problem

Test run	Weight variant[1]	Average Stiffness ratio	Casual Load (W/m^2)	Terminal unit: Set point (°C)	Terminal unit: Propor-tional Band (K)	Terminal unit: Status	CE for the following numerical methods: Alex2	BDF2	BEM	Carroll(a)[2]	Carroll(b)	IIE[2]	Kvaerno3	ROS2(a)[2]	ROS2(b)[2]	TR(a)	TR(b)[2]	TR(c)	TR-BDF2(a)	TR-BDF2(b)[2]	TR-BDF2(c)[2]	TR-BDF2(d)[2]	TRX2(a)	TRX2(b)[2]
1	Heavy	6,130	50	20	2.0	On	20.57	7.87	13.71		9.56		15.77			3.81		3.45	15.50				13.21	
2	Heavy	6,990	50	20	2.0	Off	29.42	5.68	16.72		16.62		15.20			3.70		3.31	16.48				23.60	
3	Medium	10,480	50	20	2.0	On	25.69	13.80	5.61	16.37	16.70	4.86	17.00	5.26	10.52	8.39	8.28	8.23	15.60	16.06	15.74	15.45	23.12	21.93
4	Medium	12,440	50	20	2.0	Off	36.35	18.91	9.01	30.77	33.21		31.78			10.19	10.82	11.51	31.07	29.77	27.91	29.43	43.47	37.38
5	Light[2]	3,190	50	20	2.0	On	23.54	5.14	8.61		5.72		16.26			2.58		2.91	10.48				10.99	
6	Light[2]	3,555	50	20	2.0	Off	14.02	4.53	7.75		8.20		18.78			5.87		5.14	11.67				12.87	
7	Medium[2]	9,120	70	21	3.0	On	42.24	10.31	9.04		20.41		20.75			6.73		6.93	14.56				27.96	
8	Medium[2]	10,960	70	21	3.0	Off	39.67	18.95	7.47		21.66		3.60			9.19		8.89	17.60				31.21	
9	Medium[3]	7,640	60	22	2.5	On	14.90	7.99	4.64	11.79	15.28	7.70	14.07	6.35	8.41	5.39	6.06	5.91	11.45	11.99	12.77	11.89	17.52	16.46
10	Medium[3]	9,600	60	22	2.5	Off	25.31	18.03	9.72		20.50		18.25			9.61		8.26	17.87				25.51	
11	Light[2]	2.21×10^{7}	50	20	2.0	On	16.43	10.47	11.68	12.96	7.11	3.22	9.21	3.87	1.75	2.47	2.36	3.13	12.66	5.45	6.45	5.41	10.24	16.32
12	Light[2]	2.24×10^{7}	50	20	2.0	Off	13.55	12.26	10.06		18.68		10.42			6.75		7.87	16.35				20.11	

[1]Details to be found in the file *Building types.mcd* on the attached CD ROM
[2]Not used in the quantitative assessment

Table 4.10 Geometric mean improvement in computational efficiency over TR(a)

Improvement factors for the following numerical methods								
Alex2	BDF2	BEM	Carroll(b)	Kvaerno3	TR(a)	TR(c)	TR-BDF2(a)	TRX2(a)
4.27	1.63	1.45	2.51	2.58	1.00	0.98	2.59	3.52

To gauge the significance of any one of these improvement factors in terms of its probably of occurance, the CE data (Table 4.9) for the two numerical methods are paired and the differences for the ten tests are statistically analysed in what is termed a 'paired difference experiment' [143]. The 95% confidence interval for μ_D, the mean difference in CE, is expressed in terms of $\bar{x}_D$, s_D and n_D, the sample mean, standard deviation and size respectively. It is

$$\bar{x}_D \pm t_{0.05/2} \frac{s_D}{\sqrt{n_D}} \tag{4.68}$$

where $t_{0.05/2} = 2.262$ is extracted from a table of Student's t-distribution with $n_D - 1$ degrees of freedom. Substituting the calculated values of the sample statistics for Alex2 and TR(a) for example, we obtain

$$20.63 \pm 2.262 \frac{8.85}{\sqrt{10}} = 20.63 \pm 6.33 = (14.30, 26.96)$$

So the true mean difference between the two CEs lies between 14.30 and 26.96, with 95% confidence. Since the interval falls above zero, it can be inferred that $\mu_{\text{Alex2}} - \mu_{\text{TR(a)}} > 0$; that is, the mean CE for Alex2 exceeds the mean CE for TR(a). If $t_{0.001/2} = 4.781$, the largest tabulated value, is used in Expression (4.68) the interval becomes $(7.25, 34.01)$, still well above zero; so we can assert with greater than 99.9% confidence that Alex2 is more efficient than TR(a).

Figures 4.9 and 4.10 allow a visual comparison to be made between the performance of TR(a) and that of Alex2 over part of the interval for test run three (Table 4.9). The time step (k) is variable for both reference and test solutions; output for all solutions is shown at one hour intervals.

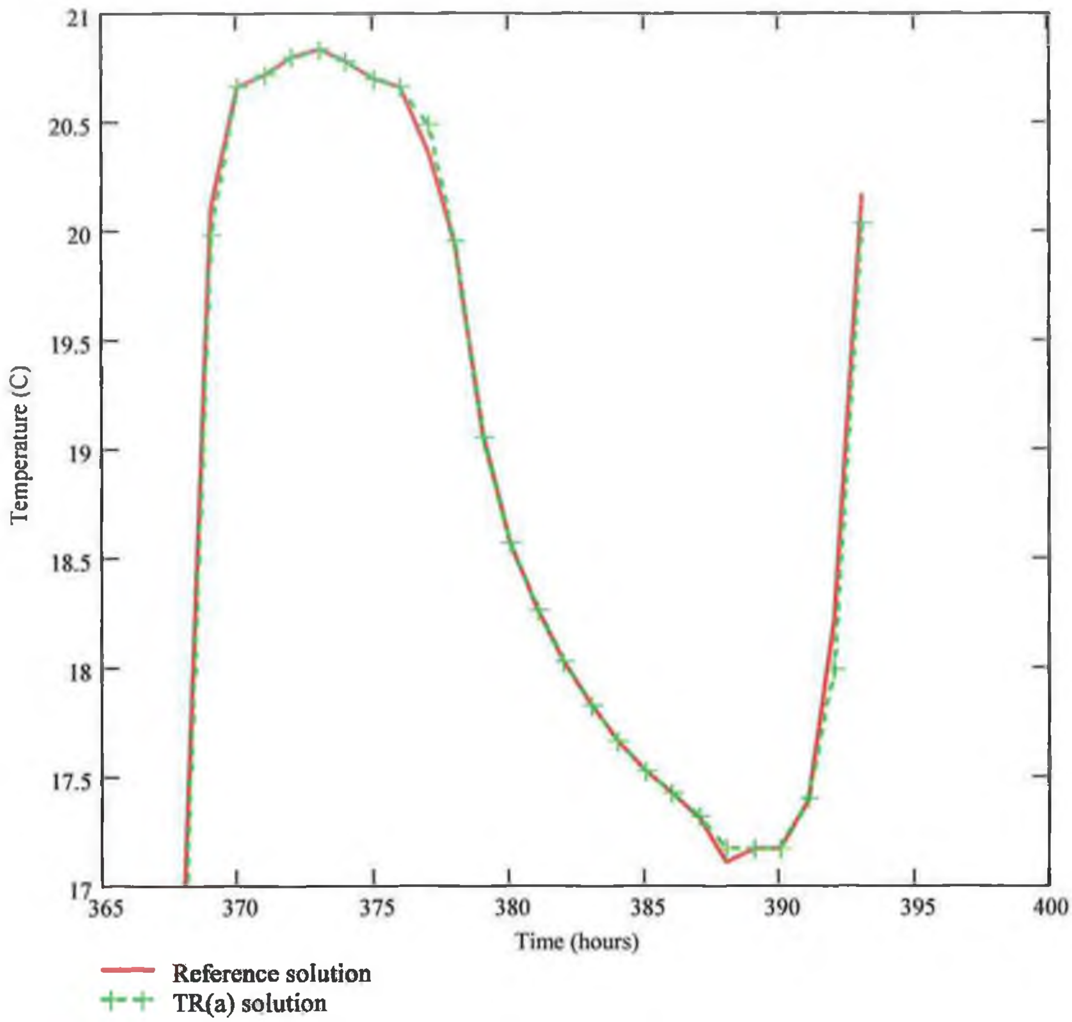


Figure 4.9 Air temperature predictions for TR(a) (Test 3, *k* variable)

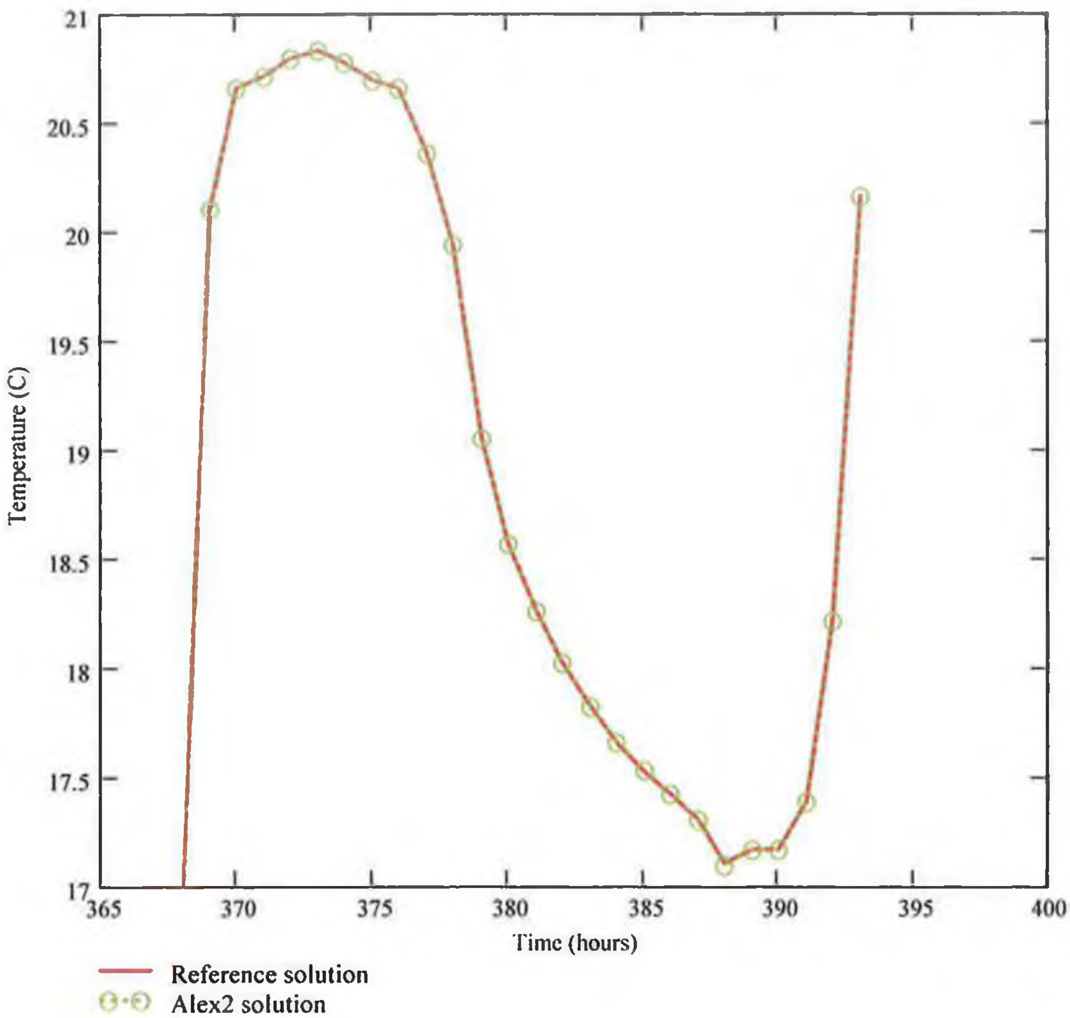


Figure 4.10 Air temperature predictions for Alex2 (Test 3, *k* variable)

4.4 Summary and Discussion

Following an introduction to stiffness in Section 4.1 and a discussion of its influence on the choice of solution method, the first of two related investigations is undertaken in Section 4.2. A direct solution method for difference equations which is used within ESP [1], the European reference model, and more widely in conduction modelling [123] is interpreted as the first iteration of an underlying iterative method. Rapid convergence of the iterative method is demonstrated. Then this method (iterated just once) is used with a simpler initial estimate to generate a novel direct solver (PM). PM was found to produce 27% less error than the commonly used method and being single-step it is also easier to implement.

The trapezoidal rule (TR) is currently used in a number of building energy simulation packages, including ESP, to solve ordinary differential equations. Because it is well-known that TR is just marginally stable for stiff systems, a range of more stable methods is introduced and assessed in Section 4.3. A short-list of the better performing methods together with their relative efficiencies is given in Table 4.10.

The performance of Kvaerno3 was disappointing considering its high order of accuracy and its resistance to order reduction. It was often the slowest to converge and it was the least tolerant of an extrapolated initial estimate. Even with a more suitable initial estimate, it sometimes produced temperature oscillations well in excess of the chosen tolerance. It should be noted that the use of Jay's device was not mentioned in Kvaerno [101]. Without it, more conservative time steps would be chosen, which would make the program more robust but less efficient.

With BDF2 it was found necessary to relax the tolerance used to terminate Newton iteration due to frequent convergence failures and this may have reduced its accuracy. Nevertheless it still generated temperature spikes at some step changes that exceeded the tolerance. This poor behaviour at discontinuities has been noted before [142] and is essentially due to the multi-step nature of BDF2. It extrapolates information from the last two solution points before the discontinuity in forming an estimate the next, but this next point in fact departs very much from the trend. All other methods examined in this work are single-step.

TR's tendency to oscillate was generally held in check by the error control routine but at the cost of smaller time steps. TR(b) and TR(c) each produced 1K temperature spikes during testing. TR(a), which includes a 'conventional' **f** evaluation, performed as well as the other two versions (perhaps unexpectedly) probably because none of the test problems was exceedingly stiff. TRX2(a) performed well, without any overt indications of its marginal A-stability. Its CE was greater than that of TR, presumably because of its smaller local error constant.

Alex2 proved the most effective numerical method for the detailed test problem. It is the optimal SDIRK formula of order two in two stages [96] but it admits no embedded third order companion formula [95] which might provide an asymptotically correct error estimate. Step-halving, used in Alexander [96] to test the method, is too expensive. The companion used here, a third order Hermite extrapolant, satisfies the second criterion (4.66) of Scraton [119] but clearly overestimates the error though not excessively so. The method offers an improvement of 327% over TR(a), the bench mark method for this study, and the statistical significance of this result is demonstrated. Also, Alex2 did not generate unrealistic temperature spikes or oscillations during testing. Of the remainder of the short-listed group, the only other methods to show evidence of spikes or oscillations were Carroll(b) and TR-BDF2(a) and they were within the tolerance of 0.1 K in both cases. None of the methods under-performed for the relatively stiff test runs 11 and 12 (Table 4.9).

Chapter 5:

Conclusions and Recommendations for Further Work

5.1 Conclusions

In Europe and the United States over 50% of all energy use can be associated with buildings and a considerable portion of this is associated with internal environmental moderation. The cost 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. Hence, a variety of building energy simulation tools are increasingly used to anticipate the thermal performance of buildings and size thermal plant. A dynamic thermal model of a building takes the form of a set of differential equations. The main goal of this work was to identify and/or develop more efficient numerical solution methods than those commonly used for the simulation of thermal energy flows in buildings.

To this end the building energy simulation problem has been characterised mathematically and a representative set of test problems formulated. These same characteristics were used to select a set of feasible numerical methods for testing. The kernel of the investigation falls into two parts; firstly, a commonly used direct solution method for difference equations was analysed and a related method developed and tested, and secondly, a collection of recently developed numerical methods for ordinary differential equations (ODE) was compared with those commonly use in this field. Testing, in both cases, was by means of numerical experiments. The tests used were realistically complex and selectively sensitive to numerical error. The chief measure of performance used was computational efficiency (CE) which is here defined to be the inverse of the product of numerical error and execution time. The latter was estimated rather than measured but this is not critical since processing time is dominated by matrix decomposition. Consequently, a correct tally on this operation alone ensures an accurate estimate of CE.

The maximum error for the direct solution method proposed in Section 4.2 has been shown to be 73% of that for a commonly used difference equation solver, by means of the test problem described in Appendix B. Since the test programs for each incorporated just one matrix inversion per time step and so incurred similar computational expense, the CE for the proposed method is greater by a factor 1/(0.731) or 37%. In the second part of the investigation, a set of numerical methods for ODEs developed in recent times has been compared with the more established methods frequently used for building energy simulation. The CE for each of the modern methods was found to be greater than that of the more frequently used ones – the most efficient being Alex2, the second-order method of Alexander [96]. Using the test problem described in Appendix C, Alex2 has been shown to be 4.27 times as efficient as the trapezoidal rule, the chosen bench mark method. Multiple tests were carried out and used to confirm the statistical significance of the result. In addition to being efficient, Alex2 proved to be very robust; it did not generate any spurious temperature spikes or oscillations during testing.

A comprehensive and discriminating test methodology has been adapted to the assessment of numerical methods for building energy simulation and is available to assess the suitability of further groups of numerical methods for this task. While it is unwise to prejudge any future candidate methods, some conclusions regarding suitable characteristics of implicit numerical methods (and associated program components) for this problem can be drawn from the work carried out to date:

(a) They are L-stable, or at least A-stable.
(b) They include a derivative function evaluation at the end of the proposed time step.
(c) They are single-step methods.
(d) They have small principal local error terms.
(e) They are of low order – probably second-order.
(f) They are used with an interval adjustment routine.
(g) They are used with an error estimator with appropriate behaviour for large time steps.
(h) The associated difference equation solver does not limit the time step.
(i) A simple initial estimate rather than an extrapolated one is used to seed the iterative difference equation solver.

It has been stated in Waters and Wright [125] and elsewhere that finite-difference schemes such as the theta method are used in many building thermal models because they are relatively simple and no single scheme is known to be superior to all others. Many other studies reviewed in Chapter 2 have also concluded that traditional methods such as BEM, TR (both emerging from the theta method) and BDF are the most appropriate for building energy simulation. In this work a number of implicit numerical methods that are appropriate to the character of the building energy problem have been identified and their efficiencies in this application quantified. The numerical method being promoted, Alex2, offers superior stability and second-order accuracy. Its computational efficiency was found to be substantially greater than that of commonly used methods for a representative test problem. It is a single-step method and therefore relatively uncomplicated to program. It is recommended for inclusion in new and existing building energy simulation software. Increased computational efficiency, coming from hardware or software improvements, can always be applied to advantage. It can be used to achieve: (i) faster simulations, (ii) greater accuracy, (iii) removal of modelling simplifications, (iv) finer sub-division of the building, or (v) enlargement of the problem domain.

5.2 Recommendations for further work

Regarding possible directions for future research, a better-matched error estimator for Alex2 would be of benefit in this application and probably many others in science and engineering. While the companion proposed here performed well, even at relatively high stiffness ratios, it does not satisfy all of the criteria considered ideal for the solution of stiff systems. Secondly, an investigation of the computational efficiency of explicit numerical methods in this field would be worth undertaking. They are probably not appropriate for the stiffer problems discussed in this work but they might prove competitive for the moderately stiff variants used for testing here.

References

1. Clarke, J.A., 1985. *Energy simulation in building design.* Bristol and Boston: Adam Hilger.

2. Clarke, J.A. and Maver, T.W., 1991. Advanced design tools for energy conscious building design: Development and design. *Building and Environment,* 26 (1), 25-34.

3. Rousseau, P.G. and Mathews, E.H., 1993. Needs and trends in integrated building and HVAC thermal design tools. *Building and Environment,* 28 (4), 439-452.

4. Cole, R.J. and Kernan, P.C., 1996. Life-cycle energy use in office buildings. *Building and Environment,* 31 (4), 307-317.

5. Negrao, C., 1995. *Conflation of computational fluid dynamics and building thermal simulation.* (PhD thesis) Glasgow: University of Strathclyde.

6. Chartered Institution of Building Services Engineers (CIBSE), 1998. *Building energy and environmental modelling.* London: CIBSE (AM 11: 1998).

7. Milbank, N.O. and Harrington-Lynn, J., 1974. Thermal response and the admittance procedure. *Building Research Series (CP 61),* 4, 205-213.

8. Carrier Air Conditioning Company, 1965. *Handbook of air conditioning system design.* New York: McGraw-Hill.

9. Stephenson, D.G. and Mitalas, G.P., 1967. Room thermal response factors. *ASHVE Trans.,* 73 no.2019.

10. Kimura, K., 1977. *Scientific basis of air conditioning.* London: Applied Science Publishers.

11. Muncey, R.W.R., 1979. *Heat transfer calculations for buildings.* London: Applied Science Publishers.

12. Clarke, J.A., 2001. *Energy simulation in building design.* (2nd ed.) Oxford: Butterworth-Heinemann.

13. Commission of the European Communities (CEC), 1990. *The PASSYS project, Phase 1: Subgroup model validation and development, Final report.* Brussels: CEC DG XII (EUR 13034 EN).

14. Holmes, M.J., 1986. The concept of the dynamic thermal model. *International Journal of Ambient Energy,* 7 (3), 117-128.

15. Pedersen, C.O., Fisher, D.E. and Liesen, R.J., 1997. A heat balance based cooling load calculation procedure. *ASHRAE Trans.*, 103 (2), 459-468.

16. Incropera, F.P. and DeWitt, D.P., 1990. *Fundamentals of heat and mass transfer.* (3rd ed.) New York: John Wiley & Sons.

17. Wiltshire, T.J. and Wright, A.J., 1988. Advances in building energy simulation in the U.K. – The Science and Engineering Research Council's programme. *Energy and Buildings,* 10 (3), 175-183.

18. Benton, R., MacArthur, J.W., Mahesh, J.K. and Cockroft, J.P., 1982. Generalized modeling and simulation software tools for building systems. *ASHRAE Trans.,* 88 (II), 838-856.

19. Rabenstein, R., 1994. Application of model reduction techniques to building energy simulation. *Solar Energy,* 53 (3), 289-299.

20. Lefebvre, G., 1997. Modal-based simulation of the thermal behaviour of a building: the m2^{m} software. *Energy and Buildings,* 25, 19-30.

21. Golten, J. and Verwer, A., 1991. *Control system design and simulation.* London: McGraw-Hill.

22. Carslaw, H.S. and Jaeger, J.C., 1959. *Conduction of heat in solids.* (2nd ed.) Oxford: Oxford University Press.

23. Wai, F.M., Letherman, K.M. and Burberry, P.J., 1982. Validation and calibration techniques for hybrid computer simulations of seasonal energy consumption. *In: Proceedings, Third International CIB Symposium on Energy Conservation in the Built Environment, Dublin, 1982.* 2.1-2.10.

24. Stevenson, W.J., 1994. Using artificial neural nets to predict building energy parameters. *ASHRAE Trans.,* 100 (2), 1081-1087.

25. Kawashima, M., Dorgan, C.E. and Mitchell, J.W., 1995. Hourly thermal load prediction for the next 24 hours by ARIMA, EWMA, LR and an artificial neural network. *ASHRAE Trans.,* 101 (1), 186-200.

26. Holland, J.H., 1975. *Adaptation in natural and artificial systems.* Ann Arbor: The University of Michigan Press.

27. Diver, D.A., 1993. Applications of genetic algorithms to the solution of ordinary differential equations. *J. Phys. A: Math. Gen.,* 26, 3503-3513.

28. Ahmad, Q.T., 1998. Validation of building thermal and energy models. *Building Serv. Eng. Res. Technol.,* 19 (2), 61-66.

29. International Energy Agency (IEA), 1994. *Empirical validation of thermal building simulation programs using test room data, Vol. 1: Final Report.* Bracknell: Building Research Establishment (IEA BCS Annex 21C/SHC Task 12 12B).

30. Parand, F., 1995. Quality assurance in the use of energy and environmental software. In: Computer modelling as a design tool for predicting building performance: Part 1. (M. Shaw ed.) *Building Serv. Eng. Res. Technol.,* 16 (4), B41-B54.

31. Lomas, K.J., 1994. Thermal program validation: The current status. *In: Proceedings, Building Environmental Performance: Facing the Future (BEP '94), University of York, April 1994.* Reading: BEPAC, 73-82.

32. Sahlin, P., 2000. The methods of 2020 for building envelope and HVAC systems simulation – will the present tools survive?. *In: Proceedings, Dublin 2000, "20 20 Vision": Conference jointly sponsored by the Chartered Institution of Building Services Engineers (CIBSE) and the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), Royal College of Surgeons, Dublin, September 2000.* London: CIBSE.

33. Hong, T., Chou, S.K. and Bong, T.Y., 2000. Building simulation: an overview of developments and information sources. *Building and Environment*, 35, 347-361.

34. Gough, M., 1999. *A review of new techniques in building energy and environmental modelling.* Watford: Building Research Establishment.

35. Crawley, D.B., 1997. Building energy tools directory. *In: Building Simulation '97, Prague, Czech Republic, Semtember, 1997.* International Building Performance Simulation Association, Vol. 1, 63-64.

36. Howard, R., Wager, D. and Winterkorn, E., 1994. *Guidance on selecting energy programs.* Cambridge: Construction Industry Computing Association.

37. Butcher, J.C., 2003. *Numerical methods for ordinary differential equations.* (2nd ed.) Chichester: Wiley.

38. Dormand, J.R., 1996. *Numerical methods for differential equations: a computational approach.* Boca Raton: CRC Press.

39. Iserles, A., 1996. *A first course in the numerical analysis of differential equations.* Cambridge: Cambridge University Press.

40. Shampine, L.F., 1994. *Numerical solution of ordinary differential equations.* New York: Chapman & Hall.

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

42. Lambert, J.D., 1991. *Numerical methods for ordinary differential systems: the initial value problem.* Chichester: John Wiley & Sons.

43. Dekker, K. and Verwer, J.G., 1984. *Stability of Runge-Kutta methods for stiff nonlinear differential equations.* Amsterdam: North-Holland.

44. May, R. and Noye, J., 1984. The numerical solution of ordinary differential equations: Initial value problems. *In*: J. Noye, ed. *Computational techniques for differential equations.* Amsterdam: North-Holland, 1-93.

45. Gear, C.W., 1971. *Numerical initial value problems in ordinary differential equations.* Englewood Cliffs, New Jersey: Prentice-Hall.

46. Integrated Environmental Solutions Ltd., 2001. *Apache simulation (APsim) calculation methods.* (Version 1) Glasgow: IES Ltd.

47. Brown, G., 1990. The BRIS simulation program for thermal design of buildings and their services. *Energy and Buildings,* 14, 385-400.

48. Wittchen, K.B., Johnsen, K. & Grau, K., 2005. *BSim User's Guide*. Danish Building Research Institute, Hørsholm, Denmark.

49. Department of Building Science, 1999. *DEROB-LTH user manual*. Sweden: Lund Institute of Technology.

50. Tang, D., 1991. The generalized system solution classes in the EKS environment. *In: Building Simulation '91, Nice, France, August, 1991*. International Building Performance Simulation Association, 323-327.

51. Charlesworth, P., Clarke, J.A., Hammond, G., Irving, A., James, K., Lee, B., Lockley, S., Mac Randal, D., Tang, D., Wiltshire, T.J. and Wright, A.J., 1991. The energy kernel system. *In: Building Simulation '91, Nice, France, August, 1991*. International Building Performance Simulation Association, 313-322.

52. Strangerup, P., 1997. *ESACAP user's manual* [online]. Available from: http://eswww.it.dtu.dk/~el/ecs/esacap.htm [Accesses 2 September 2003].

53. Lewis, P.T. and Alexander, D.K., 1990. HTB2: A flexible model for dynamic building simulation. *Building and Environment,* 25 (1), 7-16.

54. Park, C., Clarke, D.R., Kelly, G.E., 1985. An overview of HVACSIM+, a dynamic building/HVAC/control systems simulation program. *In: Building Simulation '85, Seattle, WA, August, 1985*. International Building Performance Simulation Association, 175-185.

55. Bjorsell, N., Bring, A., Eriksson, L., Grozman, P., Lindgren, M., Sahlin, P., Shapovalov, A. and Vuolle, M. 1999. IDA indoor climate and energy. *In: Building Simulation '99, Kyoto, Japan, September, 1999*. International Building Performance Simulation Association, 1035-1042.

56. Sahlin, P. and Bring, A., 1991. A tool for building and energy systems simulation. *In: Building Simulation '91, Nice, France, August, 1991*. International Building Performance Simulation Association, 339-348.

57. Eriksson, L.O., 1983. MOLCOL – an implementation of one-leg methods for partitioned stiff ODEs. Stockholm: Royal Institute of Technology. TRITA-NA-8319.

58. Lefebvre, G., Palomo, E. and Izquierdo, M., 1997. Reproducing thermal coupling between components in a generic environment like Matlab. *In: Building Simulation '97, Prague, Czech Republic, Semtember, 1997*. International Building Performance Simulation Association, Vol. 3, 7-14.

59. Jochum, P. and Kloas, M., 1993. *The dynamic simulation environment Smile* [online]. Technical University of Berlin, Institute of Energy Engineering. Available from: http://buran.fb10.tu-berlin.de/Energietechnik/ EVT_KT/smile/papers/London94.ps [Accessed 14 March 2005].

60. Lawrence Berkeley National Laboratory and Ayres Sowell Associates, Inc., 2003. *SPARK 2.0 reference manual*. California: LBNL.

61. DeLaHunt, M.J. and Palmiter, L. 1985. SUNCODE-PC: A microcomputer version of SERI/RES. *In: Building Simulation '85, Seattle, WA, August, 1985.* International Building Performance Simulation Association, 81-85.

62. Kline, S.A., Beckman, W.A. and Duffie, J.A., 1976. TRNSYS – a transient simulation program. *ASHRAE Transactions*, 82 (2), 623-633.

63. Bonin, J.L., Butto, C., Dufresne, J.L., Grandpeix, J.Y., Joly, J.L., Lahellec, A., Platel, V. and Rigal, M., 1991. The ALMETH project ZOOM code: results and perspectives. *In: Building Simulation '91, Nice, France, August, 1991.* International Building Performance Simulation Association, 355-363.

64. Cash, J.R., 2003. Review paper: Efficient numerical methods for the solution of stiff initial-value problems and differential algebraic equations. *Proc. R. Soc. Lond,* 459, 797-815.

65. Judkoff, R., Wortman, D., O'Doherty, B. and Burch, J., 1983. *A methodology for validating building energy analysis simulations.* Golden, Colorado: Solar Energy Research Institute, SERI/TR-254-1508.

66. Bloomfield, D.P., 1989. Evaluation procedures for building thermal simulation programs. *In: Building Simulation '89, Vancouver, Canada, June, 1989.* International Building Performance Simulation Association, 217-222.

67. ANSI/ASHRAE Standard 140-2001, 2001. *Standard method of test for the evaluation of building energy analysis computer programs.* Atlanta, Georgia: American Society of Heating, Refrigerating and Air-Conditioning Engineers.

68. Gough, M. and Rees, C., 2004. *Tests performed on ApacheSim in accordance with ANSI/ASHRAE Standard 140-2001.* Glasgow: Integrated Environmental Solutions Ltd.

69. Witte, M.J., Henninger, R.H., Glazer, J and Crawley, D.B., 2001. Testing and validation of a new building energy simulation program. *In: Building Simulation '01, Rio de Janeiro, Brazil, August, 2001.* International Building Performance Simulation Association, 353-359.

70. Strachan, P., 2000. *ESP-r: Summary of validation studies.* Glasgow: Energy Systems Research Unit, Univ. of Strathclyde.

71. Lomas, K.J., Eppel, H., Martin, C.J. and Bloomfield, D.P., 1997. Empirical validation of building energy simulation programs. *Energy and Buildings,* 26, 253-275.

72. Jensen, S.O., 1995. Validation of building energy simulation programs: a methodology. *Energy and Buildings,* 22, 133-144.

73. Judkoff, R., and Neymark, J., 1995. *International Energy Agency building energy simulation test (BESTEST) and diagnostic method.* Golden, Colorado: National Renewable Energy Laboratory, NREL/Tp-472-6231.

74. Editor, ECBCS News, 2003. Testing and validation of building energy simulation tools: Annex 43/SHC Task 34. *Birmingham: Energy Conservation in Buildings and Community Systems Secretariat (ESSU)*, 38, 5-7.

75. Spitler, J.D., Rees, S.J. and Xiao, D., 2001. *Development of an analytical verification test suite for whole building energy simulation programs – building fabric.* ASHRAE 1052-RP, Final Report.

76. Bland, B.H., 1992 Conduction in dynamic thermal models: Analytical tests for validation. *Building Services Engineering Research and Technology*, 13 (4), 197-208.

77. Commission of the European Communities (CEC), 1994. *The PASSYS project, Validation of building energy simulation programs, Parts 1 and 2: Research report of the subgroup model validation and development.* Brussels: CEC DG XII (EUR 15115 EN).

78. Nakhi, A.E., 1995. *Adaptive construction modelling within whole building dynamic simulation.* (PhD thesis) Glasgow: University of Strathclyde.

79. Wright, A.J., 1985 *The development and use of a model for investigating the thermal behaviour of industrial buildings.* (PhD thesis) Coventry (Lanchester) Polytechnic.

80. Underwood, C.P. and Yik, F.W.H., 2004. *Modelling methods for energy in buildings.* Oxford: Blackwell.

81. AEgis Technologies, Alabama, 2003. *ACSL SIM™ information* [online]. Available from http://www.acslsim.com/products/SIM/sim.htm [Accessed 2 September 2003].

82. Westerberg, A., 1997. *ASCEND IV: Advanced system for computations in engineering design.* [online]. Pennsylvania: Carnegie Mellon University. Available from: http://www-2.cs.cmu.edu/~ascendFTP/pdffiles/ascend-help-BOOK-3.pdf [Accessed 1 April 2005].

83. Meyer, W.S. and Liu, T., 2005. *Alternative transients program* [online]. Available from: http://www.emtp.org [Accessed 1 April 2005].

84. Nakhle, M. and Rouse, P., 1986. NEPTUNIX: An efficient tool for large scale systems simulation. *In: Proceedings, Second International Conference on System Simulation in Buildings, Liege, Belgium, December 1986.* 201-217.

85. Nagle, L. and Pederson, D.O., 1973. *Simulation program with integrated circuit emphasis (SPICE).* California: University of California, Memorandum ERL-M382.

86. Sargent, R.W.H. and Westerberg, A.W., 1964. SPEED-UP in chemical engineering design. *Trans. Inst. Chem. Eng. (London)*, 42, 190-197.

87. Barton, P. and Pantelides, C., 1994. Modeling of combined discrete/continuous processes. *AIChE J.*, 40, 966-979.

88. Petzold, L. R., 1983. *DASSL: Differential algebraic system solver.* California: Sandia National Laboratories, Category #D2A2.

89. Hairer, E. and Wanner, G., 1996. *Solving ordinary differential equations II: Stiff and differential-algebraic problems.* (2nd ed.) Berlin: Springer-Verlag.

90. Barton, P.I. and Lee, C.K., 2002. Modeling, simulation, sensitivity analysis and optimisation of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, 12 (4), 256-289.

91. Vieira, R.C. and Biscaia Jr., E.C., 2000. An overview of initialisation approaches for differential-algebraic equations. *Latin American Applied Research*, 30, 303-313.

92. Hanna, O.T., 1988. New explicit and implicit "improved Euler" methods for the integration of ordinary differential equations. *Comput. Chem.. Engng.*, 12 (11), 1083-1086.

93. Ashour, S.S., 1989. *Investigation of new methods for the integration of stiff ordinary differential equations*. (PhD thesis) Santa Barbara: University of California.

94. Bank, R.E., Coughran Jr, W.M., Fichtner, W., Grosse, E.H., Rose, D.J. and Smith, R.K., 1985. Transient simulation of silicon devices and circuits. *IEEE Trans. Comput.-Aided Design*, 4 (4), 436-451.

95. Hosea, M.E. and Shampine, L.F., 1996. Analysis and implementation of TR-BDF2. *Appl. Numer. Math.*, 20, 21-37.

96. Alexander, R., 1977. Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.*, 14 (6), 1006-1021.

97. Carroll, J., 1989. A composite integration scheme for the numerical solution of systems of ordinary differential equations. *Journal of Computational and Applied Mathematics*, 25, 1-13.

98. Carroll, J., 1993. A composite integration scheme for the numerical solution of systems of parabolic PDEs in one space dimension. *Journal of Computational and Applied Mathematics*, 46, 327-343.

99. Verwer, J.G., Spee, E.J., Blom, J.G. and Hunsdorfer, W., 1999. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM J. Sci. Comput.*, 20 (4), 1456-1480.

100. Sundnes, J., 2002. *Numerical methods for simulating the electrical activity of the heart.* (PhD thesis) Oslo: University of Oslo.

101. Kvaerno, A., 2004. Singly diagonally implicit Runge-Kutta methods with an explicit first stage. *BIT Numerical Mathematics*, 44 (3), 489-502.

102. Heath, M.T., 2002. *Scientific computing; An introductory survey.* (2nd ed.) Boston: McGraw-Hill.

103. Cash, J., 2005. *Software for initial value problems* [online]. Available from: http://www.ma.ic.ac.uk/~jcash/IVP_software/readme.php [Accessed 24 Mar 2005].

104. Mazzia, F. and Iavernaro, F., 2003. *Test set for initial value problem solvers*. Bari: University of Bari, Department of Mathematics.

105. Enright, W.H. and Pryce, J.D., 1987. Two Fortran packages for assessing initial value methods. *ACM Trans. Math. Softw.*, 13 (1), 1-27.

106. Enright, W.H. and Hull, T.E., 1976. Comparing numerical methods for the solution of stiff systems of ODEs arising in chemistry. *In*: L. Lapidus and W.E. Schiesser, eds. *Numerical methods for differential systems*. New York: Academic Press, 45-66.

107. Enright, W.H., Hull, T.E. and Lindberg, B., 1975. Comparing numerical methods for stiff systems of ODE's. *BIT*, 15, 10-48.

108. Isaacson, E. and Keller, H.B., 1966. *Analysis of numerical methods*. New York: Dover Publications.

109. Waters, J.R., 1981 An investigation of some errors due to the use of finite difference techniques for building heat transfer calculations. *Building Services Engineering Research and Technology*, 2 (1), 51-59.

110. Richtmyer, R.D. and Morton, K.W., 1967. *Difference methods for initial-value problems.* (2nd ed.) New York: John Wiley & Sons.

111. Lomas, K.J., Eppel, 1992. Sensitivity analysis techniques for building thermal simulation programs. *Energy and Buildings,* 19, 21-44.

112. Fletcher, C.A.J., 1991. *Computational techniques for fluid dynamics, Vol. 1.* (2nd ed.) Berlin: Springer-Verlag.

113. Van der Steen, A.J. and Dongarra, J.J., 2002. *Overview of recent supercomputers* [online]. Available from: http://www.phys.uu.nl/~steen/web05a/overview.html [Accessed 20 November 2002].

114. Guest, M.F., 2002. *Performance of various computers in computational chemistry* [online]. Available from: http://www.dl.ac.uk/CFS/benchmarks/compchem.html [Accessed 18 November 2002].

115. Greer, B., Harrison, J., Henry, G., Li, W. and Tang, P., 2001. Scientific computing on the Itanium processor. *In: Proceedings, 2001 ACM/IEEE conference on supercomputing, Denver, Colorado, 2001*. New York, ACM Press, 41-41.

116. Heber, G., Dolgert, A.J., Alt, M., Mazurkiewicz, K.A. and Stringer, L., 2001. Fracture mechanics on the intel itanium architecture: A case study. *In: Workshop on EPIC Architectures and Compiler Technology (ACM MICRO 34)*. Austin, Texas.

117. Lyon, T., Delano, E., McNairy, C. and Mulla, D., 2002. Data cache design considerations for the Itanium 2 processor. *In: IEEE International Conference on Computer Design: VLSI in Computers and Processors, Freiburg, Germany, September 2002.* Los Alamitos, California: IEEE, 356-362.

118. Hensen, J.L.M. and Clarke, J.A., 2000. Integrated simulation for HVAC performance prediction: State-of-the-art illustration. *In: Proceedings, Dublin 2000, "20 20 Vision": Conference jointly sponsored by the Chartered Institution of Building Services Engineers (CIBSE) and the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), Royal College of Surgeons, Dublin, September 2000.* London: CIBSE.

119. Scraton, R.E., 1986. Second-order linearly implicit methods for stiff differential equations. *Intern. J. Computer Math.*, 20, 57-66.

120. Chartered Institution of Building Services Engineers (CIBSE), 1999. *CIBSE guide A: Environmental design.* London: CIBSE.

121. Curtiss, C.F. and Hirschfelder, J.O., 1952. Integration of stiff equations. *Proc. Natl. Acad.* Sci., 38, 235-243.

122. Pinney, A. and Parand, F., 1991. The effect of computational parameters on the accuracy of results from detailed thermal simulation programs. *In: Proceedings, Building Environmental Performance (BEP '91), University of Kent, Canterbury, April 1991.* Reading: BEPAC, 207-222.

123. Ozisik, M.N., 1993. *Heat conduction.* (2nd ed.) New York: John Wiley & Sons.

124. Mathsoft, 1995. *User's guide, Mathcad PLUS 6.0.* Cambridge, Massachusetts: Mathsoft Inc.

125. Waters, J.R. and Wright, A.J., 1985. Criteria for the distribution of nodes in multi-layer walls in finite-difference thermal modelling. *Building and Environment*, 20 (3), 151-162.

126. Crandall, S.H., 1955. An optimum implicit recurrence formula for the heat conduction equation. *Q. Appl. Math.*, 13, 318-320.

127. Hensen, J.L.M. and Nakhi, A.E., 1994. Fourier and Biot numbers and the accuracy of conduction modelling. *In: Proceedings, Building Environmental Performance: Facing the Future (BEP '94), University of York, April 1994.* Reading: BEPAC, 247-256.

128. Schneider, P.J., 1955. *Conduction heat transfer.* Reading, Massachusetts: Addison-Wesley.

129. Hall, G. and Watt, J.M., 1976. *Modern numerical methods for ordinary differential equations*. Oxford: Clarendon Press.

130. Cash, J.R., 1980. A semi-implicit Runge-Kutta formula for the integration of stiff systems of ordinary differential equations. *The Chemical Engineering Journal*, 20, 219-224.

131. Rosenbrock, H.H., 1963. Some general implicit processes for the numerical solution of differential equations. *Comput. J.*, 5, 329-330.

132. Steihaug, T. and Wolfbrandt, A., 1979. A attempt to avoid exact jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Mathematics of Computation*, 33 (146), 521-534.

133. Piche, R., 1995. An L-stable Rosenbrock method for step-by-step time integration in structural dynamics. *Comput. Methods Appl. Mech. Engrg.*, 126, 343-354.

134. Zedan, H., 1990. Avoiding the exactness of the jacobian matrix in Rosenbrock formulae. *Computers Math. Applic*., 19 (2), 83-89.

135. Crowley, M.E. and Hashmi, M.S.J., 1998. Evaluation of implicit numerical methods for building energy simulation. *Proceedings of the Institution of Mechanical Engineers, Part A, Journal of Power and Energy,* 212 (A5), 331-342.

136. Gourlay, A.R., 1970. Hopscotch: A fast second-order partial differential equation solver. *J. Inst. Maths. Applics*., 6, 375-390.

137. Nevanlinna, O. and Liniger, W., 1978. Contractive methods for stiff differential equations, Part I. *BIT*, 18, 457-474.

138. Nevanlinna, O. and Liniger, W., 1979. Contractive methods for stiff differential equations, Part II. *BIT*, 19, 53-72.

139. Dahlquist, G., 1963. A special stability problem for linear multistep methods. *BIT*, 3, 27-43.

140. Jay, L., 1998. Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods. *SIAM J. Sci. Comput.*, 20, 416-446.

141. Chua, T.S. and Dew, P.M., 1981. *The simulation of a gas transmittion network using a variable-step integrator.* Leeds: Department of Computer Studies, University of Leeds. Report No. 148.

142. Byrne, G.D. and Hindmarsh, A.C., 1987. Stiff ODE solvers: A review of current and coming attractions. *Journal of Computational Physics*, 70, 1-62.

143. McClave, J.T. and Sincich, T., 2000. *Statistics*. (8^{th} ed.) New Jersey: Prentice Hall.

144. Commission of the European Communities (CEC), 1985. *Test reference years: Weather data sets for computer simulations of solar energy systems and energy consumption in buildings.* Brussels: CEC DG XII.

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

146. Berdahl, P. and Martin, M., 1984. Characteristics of infrared sky radiation in the United States. *Solar Energy*, 33 (3/4), 321-326.

147. Swinbank, W.C., 1963. Longwave radiation from clear skies. *J. R. Meteorol. Soc.*, 89, 339-348.

148. Alamdari, F. and Hammond, G.P., 1983. *Improved data correlation for buoyancy-driven convection in rooms*. Cranfield: Cranfield Institute of Technology. Report SME/J/83/01.

149. Allen, J., 1987. Convection coefficients for buildings: External surfaces. *In*: SERC/BRE, *An investigation into analytical and empirical validation techniques for dynamic thermal models of buildings*. Watford: Building Research Establishment, Vol. 2, Chapter 7.

150. Intel Corporation, 2003. *Intel Itanium 2 processor reference manual for software development and optimization* [online]. Available from: http://developer.intel.com/design/itanium/manuals.htm [Accessed July 2003]

151. Intel Corporation, 2003. *Highly optimized mathematical functions for the Intel Itanium architecture: Application note* [online]. Available from: http://www.intel.com/cd/software/products/asmo-na/eng/enabling/219871.htm [Accessed July 2003]

152. Duff, I.S. and Reid, J.K., 1996. The design of MA48: A code for the direct solution of sparse unsymmetric linear systems of equations. *A.C.M. Trans. Math. Softw.*, 22 (2), 187-226.

153. Dongarra, J.J., 2003. *Performance of various computers using standard linear equations software* [online]. Available from: http://www.netlib.org/benchmark/performance.ps [Accessed 11 March 2003].

154. McCalpin, J.D., 2003. *STREAM: Sustainable memory bandwidth in high performance computers* [online]. Available from: http://www.cs.virginia.edu/stream/ [Accessed 16 March 2003]

155. University of Melbourne, Department of Computer Science and Software Engineering, 2003. *433-313 Computer Design* [online]. Available from: http://www.cs.mu/oz.au/313/ [Accessed 26 November 2003].

156. Wallin, D., Johansson, H. and Holmgren, S., 2003. *Cache memory behaviour of advanced PDE solvers* [online]. Department of Information Technology, Upsalla University. Available from: http://www.it.uu.se/research/reports/2003-044/ [Accessed 27 November 2003].

157. Zhong, Y., Dropsho, S.G. and Ding, C., 2003. *Miss rate prediction across all program inputs* [online]. Computer Science Department, University of Rochester. Available from: http://www.cs.rochester.edu/~cding/Documents/Publications/pact03.pdf [Accessed November 2003].

158. Harrison, J., Kubaska, T., Story, S. and Tang, P.T.P., 1999. The computation of transcendental functions on the IA-64 architecture. *Intel Technology Journal*, 1999, Q4, 1-7.

Appendix A: Alternative iterative method

Another iterative solution method for Equation 4.18 can be constructed if the function $\mathbf{f}$ can be decomposed in the following way:

$$\mathbf{f}(t,\mathbf{T}) = \mathbf{G}(t,\mathbf{T})\mathbf{T} + \mathbf{g}(t,\mathbf{T}) \tag{A1}$$

where $\|\mathbf{G}\|$ is large and $\|\partial\mathbf{g}/\partial\mathbf{T}\|$ is small over the interval of interest, that is, the stiffness of $\mathbf{f}$ expresses itself in $\mathbf{G}$ rather than in $\mathbf{g}$. In this case Equation 4.18 becomes

$$\mathbf{T}^{j+1} = \mathbf{T}^{j} + \frac{k}{2}\left(\mathbf{G}^{j}\mathbf{T}^{j} + \mathbf{g}^{j} + \mathbf{G}^{j+1}\mathbf{T}^{j+1} + \mathbf{g}^{j+1}\right) \tag{A2}$$

Here superscripts have been placed on the functions to indicate the time step level. Equation A2 can be rearranged to give

$$\left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)\mathbf{T}^{j+1} = \left(\mathbf{I} + \frac{k}{2}\mathbf{G}^{j}\right)\mathbf{T}^{j} + \frac{k}{2}\left(\mathbf{g}^{j} + \mathbf{g}^{j+1}\right) \tag{A3}$$

$$\therefore \mathbf{T}^{j+1} = \left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)^{-1}\left\{\left(\mathbf{I} + \frac{k}{2}\mathbf{G}^{j}\right)\mathbf{T}^{j} + \frac{k}{2}\left(\mathbf{g}^{j} + \mathbf{g}^{j+1}\right)\right\} \tag{A4}$$

Equation A4 is another possible fixed point iteration. If its right hand side is denoted by $\mathbf{R}$ and if μ_i $(i = 1,2,\ldots,n)$ are the eigenvalues of $\partial\mathbf{R}/\partial\mathbf{T}^{j+1}$, the Jacobian matrix of $\mathbf{R}$, the condition for convergence of Equation A4 is

$$Max_i|\mu_i| = K < 1 \tag{A5}$$

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

$$\frac{\partial \mathbf{R}}{\partial T_i^{j+1}} = \left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)^{-1}\left(\frac{k}{2}\frac{\partial \mathbf{g}^{j+1}}{\partial T_i^{j+1}}\right) + \left\{-\left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)^{-1}\left(-\frac{k}{2}\frac{\partial \mathbf{G}^{j+1}}{\partial T_i^{j+1}}\right)\left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)^{-1}\right\}$$
$$\times\left\{\left(\mathbf{I} + \frac{k}{2}\mathbf{G}^{j}\right)\mathbf{T}^{j} + \frac{k}{2}\left(\mathbf{g}^{j} + \mathbf{g}^{j+1}\right)\right\} \tag{A6}$$

The identity

$$\frac{\partial\left(\mathbf{M}^{-1}\right)}{\partial T} \equiv -\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial T}\mathbf{M}^{-1} \tag{A7}$$

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

Substituting from Equation A3 for the final expression in braces, Equation A6 simplifies to

$$\frac{\partial \mathbf{R}}{\partial T_i^{j+1}} = \left(\mathbf{I} - \frac{k}{2}\mathbf{G}^{j+1}\right)^{-1}\left\{\frac{k}{2}\left(\frac{\partial \mathbf{G}^{j+1}}{\partial T_i^{j+1}}\mathbf{T}^{j+1} + \frac{\partial \mathbf{g}^{j+1}}{\partial T_i^{j+1}}\right)\right\} \tag{A8}$$

The properties of the alternative iterative method can be deduced from this last equation. Most importantly, $\mathbf{G}$ appears only in the inverted expression and, because $\|\mathbf{G}\|$ is assumed to be large (and found to be so for the systems examined), $\partial\mathbf{R}/\partial\mathbf{T}^{j+1}$ will have small elements and so $\|\partial\mathbf{R}/\partial\mathbf{T}^{j+1}\|$ will be small. Stiffness, therefore, increases the rate of convergence since $K = Max_i|\mu_i| \le \|\partial\mathbf{R}/\partial\mathbf{T}^{j+1}\|$. It can also be inferred from Equation A8 that K increases only very slowly with k when the $k\mathbf{G}^{j+1}/2$ term dominates the inverted expression. In other words large time increments do not jeopardize convergence. The convergence rate of Equation A4 is, of course, also dependent on the magnitudes of $\mathbf{T}^{j+1}$, $\partial\mathbf{g}^{j+1}/\partial T_i^{j+1}$ and $\partial\mathbf{G}^{j+1}/\partial T_i^{j+1}$ as measured by their norms.

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

(i) simple fixed point iteration [Equation 4.18],

(ii) the alternative iterative method [Equation A4] and

(iii) the Newton–Raphson process [Equation 4.21]

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

Appendix B: Simple test problem

The purpose of the test cell specification set out below is to facilitate the construction of a set of test equations with the mathematical characteristics of the building energy problem. The equations are generated by considering the heat flows at a cubic space enclosed by five identical plane slabs and one vertical glass sheet (Figure B1).

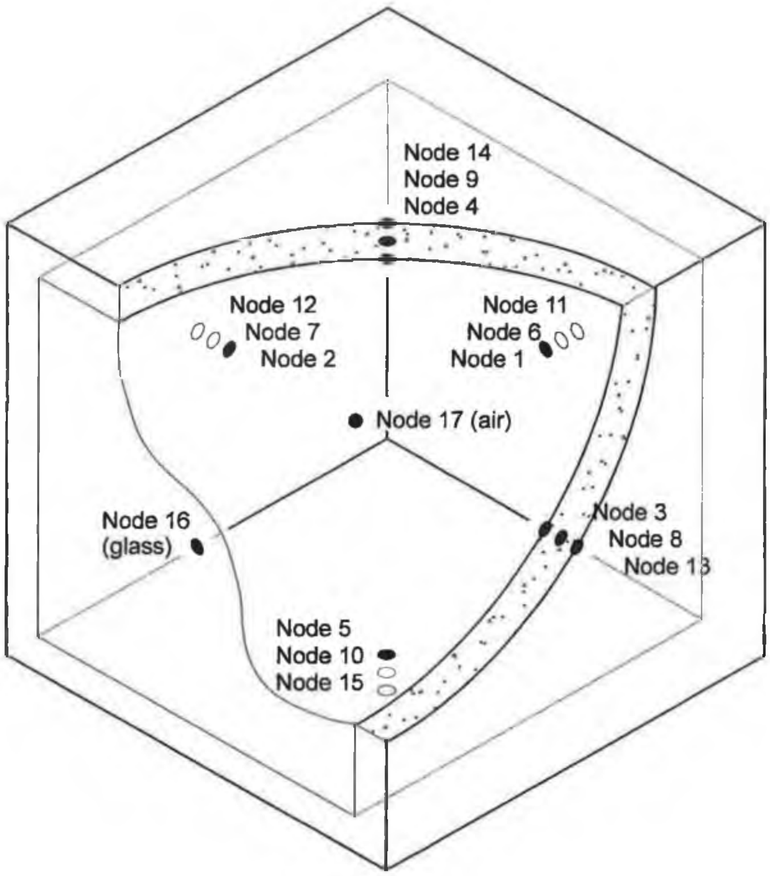


Figure B1 Test cell for simple test problem

Each three metre square slab is represented by three nodes and exchanges heat by convection with the enclosed air mass, as does the glass sheet which is represented by one node. The convection coefficient used is simple but appropriately nonlinear. Internal long-wave radiation is exchanged between opposite faces only – sufficient to introduce fourth power radiation terms to the problem. For simplicity, the emissivity is taken to be unity. External surfaces are exposed to a sinusoidally varying air temperature with a period of 24 hours, and no other thermal influence. The diurnal range of the sine wave is reduced to prevent outside air temperature dominating the other thermal loads through the large glass sheet. Short-wave radiation, entering through the glass, acts on just one internal surface – the back wall. This solar term is represented by the positive part of a 24 hour sinusoid with a 15% ripple superimposed to represent the effects of cloud. The period of the sinusoidal 'ripple' is 3.2 days approximately; less than the four day test period. The solar function includes the factors 0.5 for shading coefficient and 0.3 as glass fraction; the latter to reduce, again, the influence of the large window included in this very simple structure. A casual heat gain to the internal air mass is switched on in the morning and off again in the afternoon. A proportionally controlled convective air-conditioning terminal unit can be activated for the whole of the simulated period. Its capacity is 20% in excess of the cooling load and the proportional band of the controller is 2K.

The set of 17 differential equations, one per node, representing these interactions are set out below. Equation B0 for time $(t = T_0)$ can be included to make the set autonomous.

$$\frac{dT_0}{dt} = 1 \tag{B0}$$

$$l^2 \frac{h}{2}\rho c \frac{dT_1}{dt} = l^2 \left\{ \frac{k_s}{h}(T_6 - T_1) + h_{is}(T_{17} - T_1) + \sigma\left(T_{16}^4 - T_1^4\right) + q_{sol} \right\} \tag{B1}$$

$$l^2 \frac{h}{2}\rho c \frac{dT_i}{dt} = l^2 \left\{ \frac{k_s}{h}(T_{i+5} - T_i) + h_{is}(T_{17} - T_i) + \sigma\left(T_{i+1}^4 - T_i^4\right) \right\} \qquad i = 2,4 \tag{B2}$$

$$l^2 \frac{h}{2}\rho c \frac{dT_i}{dt} = l^2 \left\{ \frac{k_s}{h}(T_{i+5} - T_i) + h_{is}(T_{17} - T_i) + \sigma\left(T_{i-1}^4 - T_i^4\right) \right\} \qquad i = 3,5 \tag{B3}$$

$$l^2 h\rho c \frac{dT_i}{dt} = l^2 \frac{k_s}{h}(T_{i+5} - 2T_i + T_{i-5}) \qquad i = 6 \text{ to } 10 \tag{B4}$$

$$l^2 \frac{h}{2}\rho c \frac{dT_i}{dt} = l^2 \left\{ \frac{k_s}{h}(T_{i-5} - T_i) + h_{os}(T_{os} - T_i) \right\} \qquad i = 11 \text{ to } 15 \tag{B5}$$

$$l^2 d_g \rho_g c_g \frac{dT_{16}}{dt} = l^2 \left\{ h_{os}(T_{os} - T_{16}) + h_{is}(T_{17} - T_{16}) + \sigma\left(T_1^4 - T_{16}^4\right) \right\} \tag{B6}$$

$$l^3 \rho_a c_a \frac{dT_{17}}{dt} = l^2 \left\{ \sum_{j=1}^{5} h_{is}(T_j - T_{17}) + h_{is}(T_{17} - T_{16}) + q_{casual} + q_{tu} \right\} \tag{B7}$$

where:

$$h_{is} = 1.4 \left| T_{surface} - T_{17} \right|^{0.33} \qquad h_{os} = \frac{1}{0.06}$$

$q_{casual} = 50\ \mathrm{W/m^2}$ [Casual gain; eight hour square wave]

$$q_{sol} = 0.3 \times 0.5 \{1 + 0.15 \sin(\omega_{sol1} t)\} \{500 \cos(\omega_{sol2} t + \delta_{sol})\}$$

$$\omega_{sol1} = \frac{2\pi}{2.883 \times 60^2} \qquad \omega_{sol2} = \frac{2\pi}{24 \times 60^2} \qquad \delta_{sol} = -\frac{12}{24} \times 2\pi$$

[Solar ingress ($\mathrm{W/m^2}$); peaks at 12.00pm]

$$q_{tu} = -\frac{T_{17} - T_{sp}}{\mathrm{pb}/2} \times q_{tu}(\max)$$

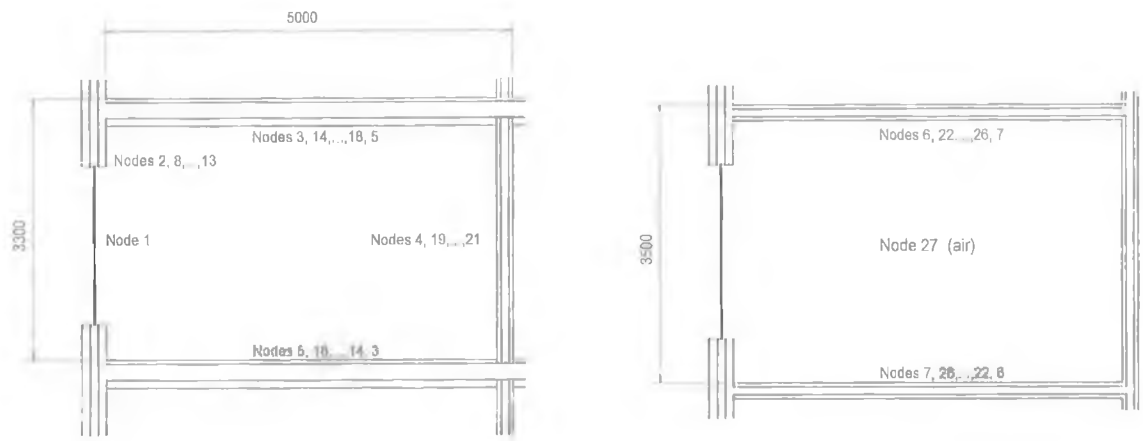
[Terminal unit output; maximum = cooling load ($\mathrm{W/m^2}$) plus 20%]

$T_{os} = 20 + 2\cos(\omega_{os} t + \delta_{os})$ [Outside air temperature (°C); peaks at 3.00pm]

$$\omega_{os} = \frac{2\pi}{24 \times 60^2} \qquad \delta_{os} = -\frac{15}{24} \times 2\pi$$

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

Appendix C: Detailed test problem



ELEVATION PLAN

Figure C1 Test room for detailed test problem

The hypothetical structure depicted in Figure C1 and specified below is used to characterise the building energy problem mathematically and to test the performance of various numerical methods in this application.

C.1 Construction details and discretization

A typical office on an intermediate floor of a large office building is used for the work. It has one external wall facing due south and, other than the single glazed window, all enclosing masonary elements are made up of three homogenous layers. The dimensional data and thermophysical properties of these layers, together with other details which vary between versions of the test, are detailed in the file *Building types.mcd* included on the attached CD ROM. Glass is represented by a single node because of its low Biot number and the small enclosed air mass is also characterised by just one node point. All other homogenous layers are represented by three nodes, one at the centre associated with half the mass and one at each surface representing a quarter of the mass in the case of a free surface or a quarter of each of two adjacent masses in the case of an interface. This is typical of building energy simulation software, as is the one dimensional representational of heat flow implied by it.

The total number of nodes required to represent the structure is apparently 44 but this can be reduced to 27 if the room is considered to be surrounded by identical rooms experiencing similar casual loads. In this case each floor/ceiling would clearly be irradiated identically (both spacially and temporally) by the sun and so the temperature distribution through each would be the same. For the same reason the temperature distribution through each side wall would be identical. If it is assumed that the internal design temperature is the same for all rooms and that solar flux never reaches the north wall, the temperature distribution through that wall would be symmetric about its centre-plane. Internal temperatures can, in fact, be expected to vary for the free running building because rooms facing north will not be directly irradiated but the assumption of uniformity is maintained to bring the computing burden within the constraints of the project. This or other minor modelling discrepancies should not invalidate the results provided all numerical methods are applied to the same problems and these problems are of an appropriate mathematical character. Figure C1 includes 27 distinct nodes numbered from inside the space to outside in all cases.

C.2 Thermal driving forces

The weather data used is taken from a test reference year (TRY) for Kew in England [144]. A TRY is composed of hourly weather data for 12 typical months, forming a year. It is used in simulating the performance of buildings and HVAC systems so that annual energy consumption, indoor comfort conditions and other quantities of interest can be estimated. Two months data were required for the present project because some of the test runs were 45 days in duration. May and June were used as these were the only consecutive months in the TRY that were taken from the same year (1963). Unrealistic discontinuities were thus avoided in progressing from one month to the next. The data is contained in the file *try03-MJ.txt* on the CD ROM. Data columns 0, 1, 2, 6, 8, 9 and 10 were used in this work and they contain dry bulb temperature (0.1°C), global radiation (J/cm^2), diffuse sky radiation (J/cm^2), wind speed (0.1 m/s), month, day and hour respectively. Where intermediate values are required, the hourly meteorological data are interpolated using cubic splines.

The impact of short wave radiation on the test room requires knowledge of solar position and intensity. Solar altitude and azimuth angles are known functions of hour and month. Direct radiation from the TRY is assigned a direction using these and its influence on any building surface quantified. An anisotropic diffuse sky model [145] is used to distribute the given diffuse radiation data because it is known to be more intense around the solar disc and at the horizon. Direct and diffuse short wave radiation, together with a ground reflected fraction (0.2) of the global radiation, are then summed for each surface of the test room. Solar transmissivity and absorptivity of glass are calculated as functions of the angle of incidence; in this context, diffuse radiation is considered to have an incidence angle of 51°, representing the average approach angle for anisotropic sky conditions [1, 13]. Self-shading by the building is considered. Solar radiation on internal surfaces is modelled as far as the first reflection.

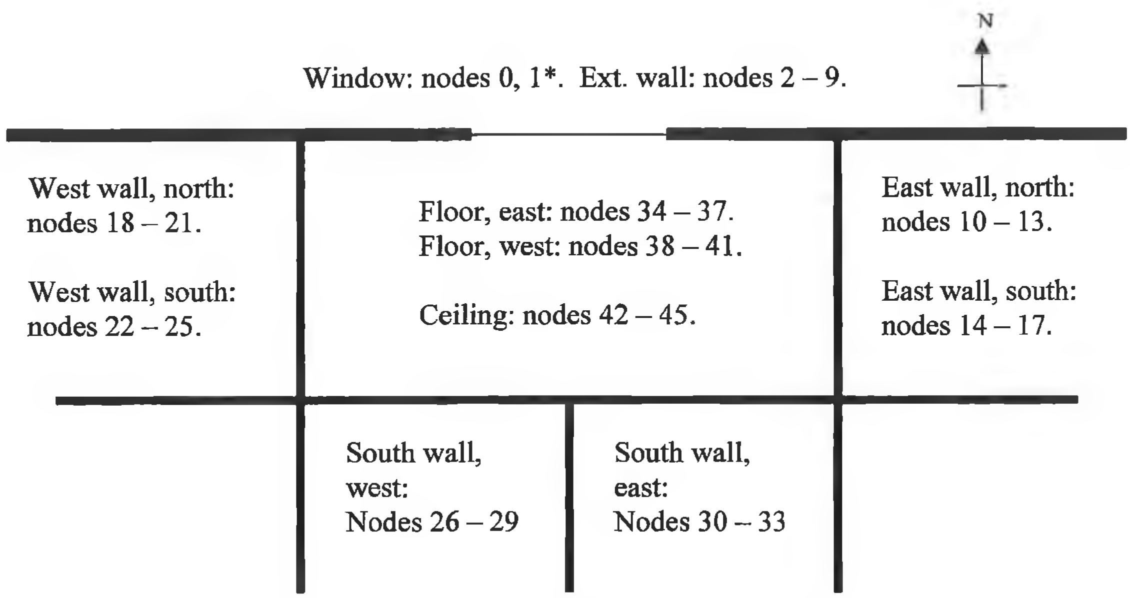
Internal long wave radiation calculations take into account multiple diffuse reflections directly between each pair of surfaces and reflections between the pair involving any third surface [1]. View factors are calculated using an area weighting method since the space approximates a cube. Net external long wave radiation exchange is estimated using a mean black body equivalent temperature of the surroundings (sky, ground and adjacent buildings) which requires calculation of the mean sky temperature as a function of time. The expression of Berdahl and Martin [146] is used here and it reduces to the clear sky formula due to Swinbank [147] when the cloud cover factor is zero. Ground temperature and that of adjacent buildings are estimated as in Clarke [1].

The formulae of Alamdari and Hammond [148] are used to calculate the natural convection coefficients for internal surfaces. To calculate the forced convection coefficient at an external surface the local wind direction and speed is first estimated using the algorithm in ESP-r and described in [13]. This is substituted into Allen's [149] expression for the surface coefficient.

Infiltration due to wind pressure and air density difference, acting on the perimeter crack around the window, is included as a set of algebraic equations which are solved at every time step. A casual heat gain to the internal air mass is switched on in the morning and off again in the afternoon. A proportionally controlled convective air-conditioning terminal unit can be activated for any desired period. Its capacity is 20% in excess of the cooling load and the proportional band of the controller is variable. The 27 differential equations representing the test room (plus one for time to make the set autonomous) are too lengthy to reproduce here and can be found in *Room acc.mcd* on the attached CD ROM.

Appendix D: System matrix for a medium-sized building

A typical, medium-sized office building of 7500 m^2 is described in this appendix and it is shown that it can be modelled by 3800 equations. Buildings of roughly this size are likely to form the majority of those requiring simulation. Since large buildings are a lot less numerous and small buildings compute very quickly anyway, a mid-range building can be justified for this study.



*Nodes for each building element are numbered from inside notional space

Figure D1 Notional space in a typical medium-sized building

The building is assumed to have six floors each of 1250 m^2. Each is considered to consist of about one-third small offices (36 x 12 m^2), one-third medium-sized spaces (6 x 70 m^2) such as lobbies or large offices and one-third large spaces (1 x 400 m^2), for example, an open-plan office or a lecture room. It is assumed that the small offices divide into six groups and within each the offices are identical. The number of different spaces to be considered on each floor is then 13 and the aspect ratio of the floors is taken to be between 2:1 and 3:1. The total number of spaces in the building is 6 x 13 = 78.

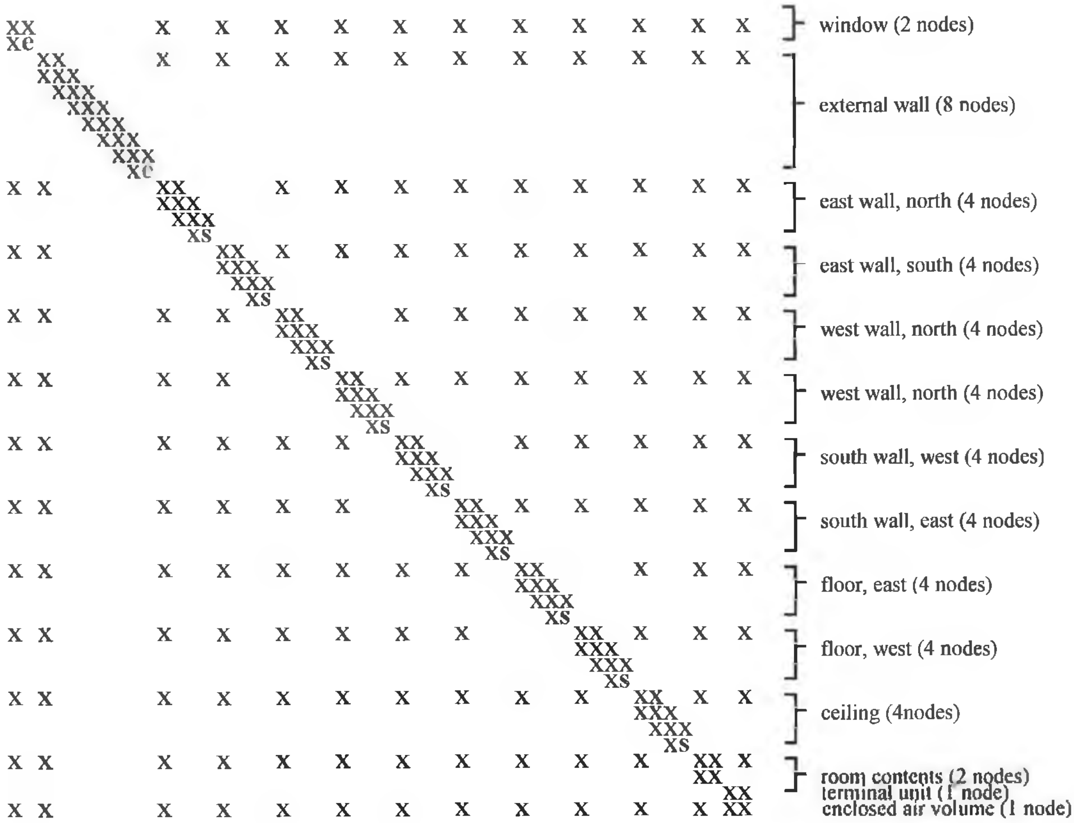


Figure D2 System matrix (50 x 50) for the notional space of Figure D1

A notional space in the office building is represented in Figure D1. It is bounded by six planes, one of which will likely be an external wall in a building of this size and shape. The external wall and window cannot exchange long wave radiation and it is assumed that four of the remaining five building elements (excepting the ceiling, say) are also divided into coplanar pairs each with zero relative view factors. This is clearly necessary in the case of the south wall and is done to improve modelling accuracy (say) in the case of the other surfaces. A typical building element is represented by approximately eight nodes and half of these are associated with the room in question for all except the external wall which is represented in its entirety within this room model.

The non-zero elements of the system matrix are shown in Figure D2 and these signify thermal coupling between the nodes of the notional space. In addition, some entries are represented differently indicating that these nodes also interact with nodes outside the space in question. The two elements marked ‘e’ are coupled to the external surface node of a space on the south face of the building immediately behind the notional space. This node, of all those within the system, best approximates the thermal conditions on the south face of the building opposite with which the elements marked 'e' actually exchange heat. The nine elements marked 's' are coupled to adjacent spaces through the centres of common walls.

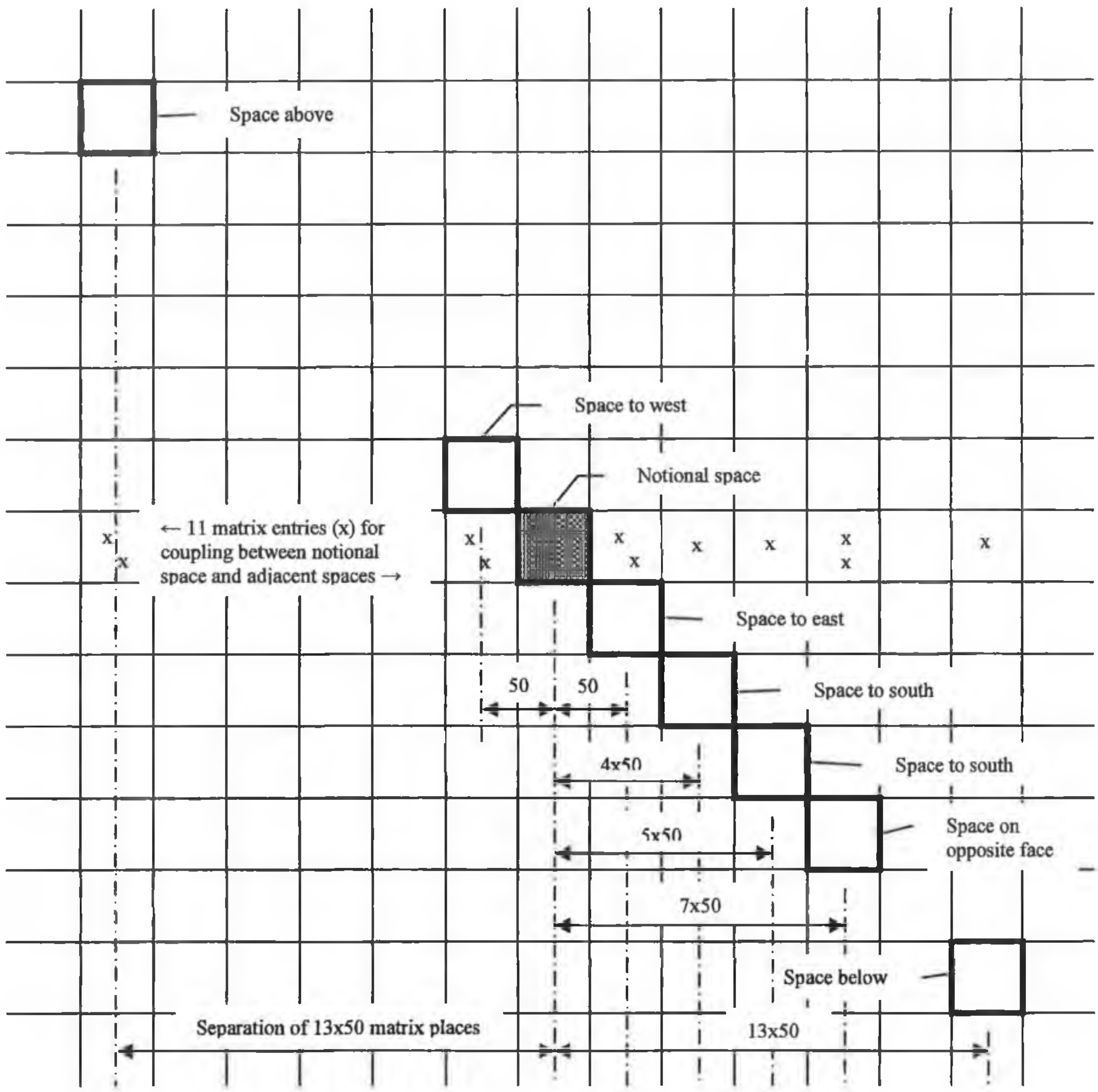


Figure D3 Part of the system matrix for a typical medium-sized building

Figure D3 represents part of the system matrix for the 7500 m^2 building. Nodes are assumed to be numbered from west to east and then from north to south on each floor, and from top to bottom of the building. As can be seen from the indicated separations (in terms of matrix entry positions) between sub-matrices representing rooms, the main diagonal has been compressed so that all spaces interacting with the notional space can be included. The only non-zero matrix elements shown, outside of the diagonal blocks, are those for heat flow to the notional space. The structure of the building matrix is, broadly speaking, block diagonal with a number of parallel, diagonal lines. In practice, the lines are not perfectly straight but include steps and gaps because the building is not perfectly regular. The total number of nodes in the building is 78 x 50 = 3900.

Appendix E: Computational effort for building energy problem

A typical, medium-sized building of 7500 m^2 floor area is described in Appendix D and it is shown that it can be modelled by 3800 equations. The significant elements of computational work required to solve the mathematical model representing this building are examined here and execution times for a single application of each on a workstation are estimated. Performance data for a Hewlett-Packard RX2600 workstation with an Intel Itanium 2 processor operating at 1 GHz are used. During a typical test run the frequency of each of the significant computational tasks is recorded. They are finally multiplied by their respective execution times and accumulated to give the expected computational effort for the 7500 m^2 building.

The tasks chosen for examination are those normally counted in numerical experiments. They are LU decomposition (factorisation), forward/back substitution, matrix evaluation and derivative function evaluation. Each requires at least $O(n)$ operations, where n is the number of equations. In the past, it was felt sufficient to keep a tally of floating point functions and operations, and this together with a unit time for each was used to estimate computational effort. In recent times, however, processor performance has been increasing much more rapidly than that of random access memory (RAM); for example, for the Itanium 2, a basic arithmetical operation takes 5 clock cycles or less and a reference to RAM about 200 [150, 151]. To address this bottleneck, small, fast cache memories have been interposed between the processor and main memory (RAM). Their purpose is to retain data and instructions that are frequently used or that may be used in the near future. The Itanium 2 has three levels of cache, the smallest and fastest being closest to the processor. Memory performance, therefore, is beginning to dominate computational effort, making it increasingly difficult to estimate. The approach taken in this project was to use published timings where available.

E.1 Factorisation and solve

The most time consuming task is expected to be LU factorisation and with this will be considered the subsequent two triangular solves (forward and back substitution). Solution methods for linear systems can be classified as direct or iterative. Iterative methods usually require less work if convergence is rapid and they are best applied to large, regular, three-dimensional problems. Building energy simulation throws up matrices that are nonsymmetric, ill-conditioned and not always diagonally dominant. For these, iterative methods tend to be significantly less reliable [102]. Modified Newton iteration, used here, leads to the solution of systems with multiple right hand sides and this again points to the use of direct methods. Consequently, published timings for MA48, a state of the art direct solver for large, sparse, unsymmetric, linear systems, were used [152]. Eight test matrices were selected from Table 1 of Duff and Reid [152] for their similarity to the building matrix of Appendix D in that they were real, sparse, unsymmetric and dense around the diagonal. Their *nnz/n* ratios (i.e. number of non-zero elements divided by matrix order) were also close to that of the building matrix which, from Appendix D, is 78 x (270 + 11) / (78 x 50) = 5.62. The chosen matrices were cases 2, 6, 8, 9, 11, 13, 15 and 17. Timings from Table 2 of Duff and Reid [152] for 'first factorise' on one processor of a Cray YMP-8I/8128 were fitted with a function of the form $an^2 + b$ which was then extrapolated to $n = 3900$ (the order of the building matrix of Appendix D) to give 2.66 seconds. This was repeated for 'solve' timings (taken from the same table and including both forward and back solves) using the fitting function $an^{4/3} + b$ to give 0.0198 seconds for $n = 3900$. The fitting functions used reflect the expected scaling for sparse, direct methods applied to a regular, three-dimensional problem. The building energy problem presents a somewhat less regular matrix but the fitting functions are, nevertheless, considered the best available for the purpose. Timings from Table 4 of Duff and Reid [152] for an IBM RS6000, model 550 were processed in a similar way to give 5.48 s and 0.0676 s for 'factorise' and 'solve' respectively. Since Duff and Reid (1996) did not present results for a HP RX2600 Itanium 2, the extrapolated execution times above were scaled to give the expected times for the HP workstation using performance data collated by Dongarra [153]. The figures quoted there are processing rates for a wide range of computer systems for the solution of a linear system of order 1000. They are close to the peak performance for the processor in each case, as are the results for MA48. They are: HP RX2600, 3528 Mflops/s; Cray YMP, 313 Mflops/s and IBM RS6000, 70 Mflops/s. The Cray times were scaled by the factor 313/3528 and the IBM times by 70/3528 before the results were averaged (in pairs) to give the final estimated timings for the Hewlett-Packard RX2600 workstation as 'factorise', 0.173 s and 'solve', 1.55×10^{-3} s.

E.2 Matrix and function evaluation

The related tasks of evaluating the building matrix (the jacobian **J** for the most part) and the derivative function **f** have little regularity and so the performance data of Dongarra [153] are not applicable. The task on which these data are based involves a high degree of data re-use.

Typically, the matrix is decomposed into b x b blocks, where b is determined by the cache size, and $O(b^3)$ operations are performed on each block before the next is loaded. This recent optimisation is in response to the 'memory bottleneck'. Neither can the performance data of McCalpin [154] be used. They are measures of sustainable memory bandwidth for many computer systems and are specifically intended to eliminate the possibility of data re-use. They are found by presenting regularly arrayed data (long vectors) to the machine for simple processing, such as copying or adding. The evaluations of **f** and **J** offer little scope for data re-use or the 'streaming' of data to/from memory. Instead, each of the subfunctions [e.g. I_t( ), incid( ), h_{nc}( )] listed in *Room acc.mcd* on the attached CD ROM, that make up the components of **f** and **J** for the detailed test problem of Appendix C, was further analysed into its basic mathematical functions and operations [e.g. tan, exp, division, fma (fused multiply-accumulate)]. Timings for these fundamental functions are available [151] and the frequency of the subfunction calls is easily established. For example, incid( ), the solar incidence angle, is evaluated once per façade per time step and h_{nc}( ), the convection heat transfer coefficient, is evaluated once per internal surface per time step.

Memory access times (hit times) are also readily available for the Itanium 2 processor [150]. It has three levels of cache, all mounted on the chip, the nearest to the arithmetic and logic unit (ALU) being L1. Hit times for the various memory levels are as follows:

Table E1 Memory access times for the Itanium 2

Memory level	Hit times (clock cycles or 10^{-9} s) Floating point data	Instructions
L1	*	1
L2	6	13
L3	15	15
RAM	225	225

* Floating point data bypasses L1

The difficulty in using these figures is that it is not known where the data and/or instructions for the next function or operation reside. The broad strategy used by programmers for cache management is to ensure that recently addressed information together with adjacent information (the next block of instructions or array elements) is available in the caches. Generally, the more locality (of either kind), the higher the hit rate. The concept of mean access time (MAT) is used here to allow an estimate of memory use time to be added to the execution times for basic mathematical functions and operations [155].

$$\text{MAT} = \text{hit time} + \text{miss rate x miss penalty} \quad \text{(E1)}$$

where the 'miss penalty' is the time to look up the data in the next (slower) level of memory plus the time to transfer it to the faster level (loading time).

Miss rates for a wide range of advanced engineering programs have been measured by Wallin *et al* [156] and simulated by Zhong *et al* [157] with consistent results which were averaged for use in this project. The averages are (a) 3.9% between L1 and L2, (b) 2.3% between L2 and L3 and (c) 2.3% between L3 and RAM. Results (b) and (c) are the same because the cache line size is 128 Bytes for both L2 and L3 whereas for L1 it is 64 B. Loading time is just load size divided by loading rate and the latter is 32 B per clock cycle for all memory levels [117]. The load size is 8 B for a floating-point number and approximately 77 B for an instruction. The second figure is justified as follows. The largest of the basic mathematical functions used here are sin and cos, both of which make use of a common 768 B table [158]. The other functions requiring the loading of a table are rarely called from **f** or **J**. Division requires about ten program lines at perhaps 5 B per line. Most other basic functions require much less memory space or are rarely used here. Addition, subtraction and multiplication are hardware operations. By inspection, about 10% of the basic function calls are to sin, cos or divide, so approximately 768/10 = 77 Bytes of instructions must be loaded for an average basic function call. This is an estimate but subsequent calculations will show that MAT is not a sensitive function of load size. Finally, the loading time for floating-point data is 8/32 = 0.25 cycles and for instructions is 77/32 = 2.4 cycles; each clock cycle having a duration of 10^{-9} seconds.

Equation E1 is next used repeatedly to find the MAT for data (D) and instructions (I) between successive pairs of memory levels.

Between L3 and RAM:

$$\text{MAT}_\text{D} = 15 + 0.023(225 + 0.25) = 20.2 \text{ cycles} \quad \text{(E2)}$$

$$\text{MAT}_\text{I} = 15 + 0.023(225 + 2.4) = 20.2 \text{ cycles} \quad \text{(E3)}$$

Between L2 and L3:

$$\text{MAT}_\text{D} = 6 + 0.023(20.2 + 0.25) = 6.5 \text{ cycles} \quad \text{(E4)}$$

$$MAT_I = 13 + 0.023(20.2 + 2.4) = 13.5 \text{ cycles} \tag{E5}$$

Between L1 and L2:

$$MAT_I = 1 + 0.039(13.5 + 2.4) = 1.6 \text{ cycles} \tag{E6}$$

The mean access times for floating-point data and instructions are 6.5 cycles and 1.6 cycles respectively, and these together with timings for the basic mathematical functions [151] were used to estimate evaluation times for **f** and **J**. Some subfunctions are called by both **f** and **J**. Their cost is included in the estimate for **f** but not that of **J** because **f** is evaluated far more often than **J** and whenever **J** is evaluated **f** is also evaluated in the same time step.

Table E2 Computational effort for the most demanding tasks

Computational task	Execution time (s)
LU factorisation (decomposition)	1.73×10^{-1}
Triangular solves (Forward/back substitutions)	1.55×10^{-3}
Matrix evaluation	1.11×10^{-3}
Derivative function evaluation	1.07×10^{-3}

The final timings for the major computational tasks are presented in Table E2. LU decomposition (factorisation) of the system matrix is the dominant computational task, even when the frequencies of the various linear algebra operations are factored in (see Table 4.8).

Appendix F: Published work

Papers in Refereed Journals

Crowley, M.E. and Hashmi, M.S.J., 1998. Evaluation of implicit numerical methods for building energy simulation. *Proceedings of the Institution of Mechanical Engineers, Part A, Journal of Power and Energy,* 212 (A5), 331-342.

Crowley, M.E. and Hashmi, M.S.J., 2000. Improved direct solver for building energy simulation. *Proceedings of the Chartered Institution of Building Services Engineers, Series A, Building Services Engineering Research and Technology,* 21 (3), 169-175.

Refereed Conference papers

Crowley, M.E., Hashmi, M.S.J., 1998. Analysis of quasi-linear solution method in Environmental Systems Performance and presentation of a new related method. *In: 3rd International Congress on Heating and Air Conditioning of Buildings: Energy and Environment, Maribor, Slovenia, May 1998.* Slovenia: University of Maribor, 221-233.

Crowley, M.E., Hashmi, M.S.J., 1998. Evaluation of implicit numerical methods for building energy simulation. *In: 2nd European Conference on Energy Performance and Indoor Climate in Buildings* and *3rd International Conference on Indoor Air Quality, Ventilation and Energy Conservation in Buildings, Lyon, France, November 1998.* Lyon: Ecole Nationale des Travaux Publics de l'Etat, 831-836.

Crowley, M.E., Hashmi, M.S.J., 2000. Development of exponential-fitted numerical methods for building energy simulation. *In: Dublin 2000, "20 20 Vision": Conference jointly sponsored by the Chartered Institution of Building Services Engineers (CIBSE) and the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE), Royal College of Surgeons, Dublin, September 2000.* London, CIBSE.

Conference Papers

Crowley, M.E., 1999. Computer simulation in building services engineering and related disciplines. *Paper delivered to The Chartered Institution of Building Services Engineers (Republic of Ireland Branch), IEI, Clyde Road, Dublin, February 1999.*

Crowley, M.E., Hashmi, M.S.J., 2001. Development of a direct solution method for building energy simulation. *In: 8th Annual Symposium of the Irish Society for Scientific and Engineering Computation, University College Dublin, May 2001.* Dublin, ISSEC.

Crowley, M.E., Hashmi, M.S.J., 2002. Efficient, implicit solvers for building energy simulation. *In: 9th Annual Symposium of the Irish Society for Scientific and Engineering Computation, University College Galway, May 2002.* Dublin, ISSEC.

Crowley, M.E., 2002. Development of an improved direct solver for building energy simulation. *Annual Patrick Benson Memorial Lecture, D.I.T. Bolton St., Dublin, November 2002.*

Crowley, M.E., 2005 Numerical methods for building energy simulation. *Energy and Buildings Seminar, Focas Institute, DIT, April 2005.*

Articles in Professional Magazines

Crowley, M.E., Costelloe, B. and Demetriou, L., 2001. Dublin Institute of Technology. *BEPAC Research Update*, 3, 47-48.

Appendix G: Attached CD ROM

The files and programs used to test numerical methods in this project are included in two formats on the attached CD ROM. Those within the folder *Mathcad* can be run and amended within Mathcad 2000 Professional and the copies in the folder *Acrobat* are 'read only' and were created using Acrobat 5.0. Subfolders within these folders contain the files associated with the investigations described in Sections 4.2 and 4.3. The following notational switches should be noted in moving from the dissertation text to the program files:

$\mathbf{T} \rightarrow \mathrm{y}$ $\quad$ $\mathbf{f} \rightarrow \mathrm{D}$ $\quad$ $\mathbf{G} \rightarrow \mathrm{M}$ $\quad$ $\mathbf{g} \rightarrow \mathrm{c}$ $\quad$ $j \rightarrow \mathrm{r}$ $\quad$ $S \rightarrow \sigma$

G.1 An Improved Direct Solution Method

Eigenvalues for the simple test problem (Appendix B) are computed in *Cube_eig.mcd* and then used to calculate the stiffness ratio, the pre-conditioning period and the terminal unit (FCU) output. *Cube_acc.mcd* was used to generate accurate (converged) solutions to the variants of the test problem and these are included as files of the form *zt_*.prn* and *zzt_*.prn*. The programs *TR+*.mcd* are used to test four direct (difference equation) solvers. They compute and evaluate test solutions quickly and so the results are not saved to disc. The properties of the construction materials used in the various tests are to be found in *Materials.mcd* in a form suitable for copying and pasting into the programs above. Finally, geometric means are calculated in the program *GM.mcd*.

G.2 Evaluation of Implicit Numerical Methods

Eigenvalues for the detailed test problem (Appendix C) are computed in *Room eig.mcd* and then used to calculate the stiffness ratio, the pre-conditioning period, the time 'gap' between discontinuities and the consequent peak disturbance and, finally, the terminal unit (FCU) output. *Room acc.mcd* was used to generate accurate (converged) solutions to the variants of the test problem and these are included as files of the form *zt;*;*.prn* and *zzt;*;*.prn*. The numerical methods for testing are in the subfolder *Methods* and they are copied and pasted into *Run, save & test; METHOD.mcd* to produce a result which is saved in the subfolder *Results*. The results are examined using *Load & test; RUN.mcd*. Since the test variant *light[2] wt* has fewer nodes than any of the others, a parallel set of programs with the italisized term in their titles was included to process this test. The program *Load & compare; 2 RUNS; light wt & light[2] wt.mcd* allows results for two versions of the lightweight building to be compared. The files *Perez.prn* and *try03-MJ.txt* are for use within the above programs. The properties of the construction materials and the values of other quantities which vary from test to test variants are to be found in *Building types.mcd* in a form suitable for copying and pasting into the programs above. Finally, geometric means are calculated in a program of that name.

Programs in the folder *Diagonal dominance* investigate three models for this property, (i) the detailed test problem (*Room*), (ii) the simple test problem (*Cube*) and (iii) an even simpler room model (*Dtm*) originally designed to examine control modes. The system matrices for various versions of the third model do not possess this property so diagonal dominance cannot be assumed.