Hopscotch: a Fast Second-order Partial Differential Equation Solver

A. R. Gourlay†

Department of Mathematics, University of Dundee, Dundee, Scotland

[Received 30 September 1969]

An idea of Gordon for the numerical solution of evolutionary problems is reformulated and shown to be equivalent to a Peaceman–Rachford process. A fast computational process is then developed and applied to parabolic and elliptic problems, both linear and non-linear. This algorithm is very efficient with regard to computing time, storage requirements and ease of programming. Several fairly general conditions are given which ensure convergence for parabolic and elliptic problems.

1. Introduction

The numerical solution of first and second order partial differential equations poses a difficult problem for the numerical analyst and, as yet, no completely reliable algorithm exists which is computationally efficient. In particular, the solution of elliptic systems has received considerable attention and many of the most commonly used methods are discussed in the books by Varga (1962) and Wachspress (1966). These methods include the almost classical successive overrelaxation (which is perhaps still most frequently used) and the more recent alternating direction iteration (which, by itself, is not suited to general or “difficult” problems). More recently, an interesting algorithm which appears well-suited to difficult elliptic problems has been proposed and analysed by Stone (1968), Dupont, Kendall & Rachford (1968) and Dupont (1968). In the case of a parabolic problem (or indeed most evolutionary problems) the most successful procedure to date appears to be based on alternating direction implicit algorithms (or, more generally, on the class of fractional-step methods). Such algorithms generate difficulties within themselves as the number of space dimensions is increased. In an attempt to alleviate these difficulties, there has been widely proposed in the Russian literature, the class of locally one-dimensional (L.O.D.) methods, recently shown to be equivalent, in a certain sense, to the class of A.D.I. methods, Gourlay & Mitchell (1969). However, L.O.D. methods often suffer severe loss of accuracy and this is shown in examples quoted later. The philosophy behind A.D.I. theory is that of breaking the problem up into a series of simpler computational problems. Usually one generates a simpler problem corresponding to each of the space dimensions. This process may require considerable computation with problems in many space dimensions and may also produce storage difficulties. Moreover, in the case of non-linear problems the complexity of the algorithm may be increased considerably although, in the elliptic case, the inner-outer iteration algorithm of Gunn (1964) appears to be effective.

In this paper, we propose an algorithm for the solution of such partial differential equations.

† Present address: IBM (United Kingdom) Ltd., Peterlee, Co. Durham, England.
equations which enjoys ease of use, efficiency in machine time and economy of storage but, as yet, lacks a complete theory. However, the limited experience of the author in using this technique appears to warrant its further study. It is hoped that, by publishing the results so far obtained and demonstrating the power of the technique, an improved algorithm may result.

It is stressed at this stage that the motivation for this algorithm is the paper by Gordon (1965) and two related papers by Scala & Gordon (1966, 1968) wherein the idea of using a varying scheme to calculate the finite difference solution to the problem is proposed. The analysis of Gordon (1965) is necessarily limited by his approach. The present analysis is motivated by showing that the algorithm may be regarded as an A.D.I. method with a rather novel method of decomposing the problem into its two simpler parts. The algorithm proposed in this paper always leads to a two-stage process independent of the number of space dimensions.

The outline of the paper is as follows. In Section 2 we introduce the algorithm of Gordon, redefine it locally, thus giving a fast computing algorithm, and then consider it globally. In Section 3, we study the solution of linear parabolic problems, prove stability and give a weak estimate of rate of convergence. In Section 4, the corresponding elliptic problem is considered and the convergence of the sequence of iterates is proved. In Section 5, we show how non-linear problems may be handled with little additional effort. In Section 6, the extension of the algorithm to many space dimensions, its efficiency in machine time and storage requirements and its general applicability are considered. Finally, in Section 7, some numerical results and comparisons are given.

In the paper of Gordon (1965), the original technique was described as "a non-symmetric difference equation". The present author feels that in fact the algorithm enjoys a considerable degree of symmetry and, therefore, that this name is inappropriate. In discussions with colleagues, the algorithm was referred to as the "hopscotch" process because its progress through the space-time grid resembles this game. In what follows, therefore, we shall refer to the new algorithm as the hopscotch process.

2. The Hopscotch Algorithm

In order to develop the algorithm we restrict ourselves for the moment to the linear equation

$$\frac{\partial U}{\partial t} = LU + g(x, y, t)$$  \hspace{1cm} (2.1)

where $L$ is a second-order linear, elliptic differential operator in the space variables $x$ and $y$. The solution is required in the cylinder $R \times [0 \leq t \leq T]$ where $R$ is a closed region in the $x$-$y$ plane, with continuous boundary $\partial R$. Appropriate initial and boundary data are given on $t = 0$ and $\partial R \times [0 \leq t \leq T]$ respectively.

We superimpose a square grid on the region $R$, giving the set of points $(ih, jh) \in R_h$, where $i, j$ are integers and require the solution to (2.1) at the grid points in $R_h \times \{t_m\}$, $m = 1, \ldots, M$, where $t_m = mk$ is a plane parallel to $R$, with $k$ and $h$ the mesh spacings in time and space respectively. We let $u^n_{ij}$ denote the approximate solution to equation (2.1) at the point $(ih, jh, mk)$. The exact solution of the differential equation at this point is $U(ih, jh, mk)$. 

The algorithm of Gordon (1965) consists of using alternately the simple explicit and implicit replacements of equation (2.1) namely

\[ u_{ij}^{n+1} = u_{ij}^n + k(L_h u_{ij}^n + g_{ij}^n) \] (2.2)
\[ u_{ij}^{n+1} = u_{ij}^n + k(L_h u_{ij}^{n+1} + g_{ij}^{n+1}) \] (2.3)

where \( L_h \) is a finite difference replacement of the linear operator \( L \). To evaluate the solution at the points \((i, j, m+1)\) Gordon uses first of all equation (2.2) for those points which have \((i+j)\) even and then using (2.3) at those points for which \((i+j)\) is odd. If \( L_h u_{ij}^{n+1} \) is a replacement which involves \( u_{ij}^{n+1} \) and its nearest neighbours along grid lines (i.e. \( u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1} \)) only then this algorithm is explicit. We shall henceforth assume that \( L_h \) has this property. (We shall refer to such an operator as an \( E \)-operator. An \( E \)-operator can only be a replacement of a first or second order differential expression. Such operators are, however, abundant in numerical analysis and it follows that the hopscotch process is widely applicable in its explicit form. If \( L_h \) is not an \( E \)-operator, then the hopscotch procedure is implicitly defined. We return to this point in Section 6. The \( E \)-operator property is, in fact, a powerful feature in the algorithm.)

At the following time step, the roles of odd and even formulas are interchanged. Therefore, equation (2.2) is used at points for which \((i+j)\) is odd and (2.3) is used at the points with \((i+j)\) even. Gordon proves the convergence of this algorithm under the restriction \( k/h \to 0 \) as \( k, h \to 0 \). We use alternative methods of proof based on the following formalism and obtain more general results. In addition, we apply the process to elliptic problems and prove convergence.

Our formulation consists of replacing the two equations (2.2) and (2.3) by a single equation which defines the algorithm locally at all points. We introduce the following mesh function (the odd–even function)

\[ \theta_{ij}^n = \begin{cases} 1 & \text{if } m+i+j \text{ is odd} \\ 0 & \text{if } m+i+j \text{ is even} \end{cases} \]

and then define our replacement of (2.1) to be the following equation

\[ u_{ij}^{m+1} - k\theta_{ij}^{m+1}[L_h u_{ij}^{m+1} + g_{ij}^{m+1}] = u_{ij}^m + k\theta_{ij}^m[L_h u_{ij}^m + g_{ij}^m]. \] (2.4)

We notice that since our algorithm changes over at succeeding steps we really only obtain "answers" at \( m+1 = 2n \) when \( n = 0, 1, \ldots \). However, since the solution at odd values of \((m+1)\) are meaningful approximations to the solution of the differential equation at \( t = (m+1)k \), we may use them as such. {This situation is comparable to that which occurs if we use the standard Peaceman–Rachford method. In locally one-dimensional methods, no such meaning can be applied to intermediate solutions. A fuller discussion of the merit of having consistency at the half-step in connection with the Peaceman–Rachford and L.O.D. methods is given in Gourlay & Mitchell (1969).} This is the reason why we regard (2.4) as advancing the calculation from \( t = mk \) to \( t = (m+1)k \). But the merit of the formulation (2.4) first appears if we write down two successive equations of type (2.4) namely

\[ u_{ij}^{m+1} - k\theta_{ij}^{m+1}[L_h u_{ij}^{m+1} + g_{ij}^{m+1}] = u_{ij}^m + k\theta_{ij}^m[L_h u_{ij}^m + g_{ij}^m], \] (2.4a)

\[ u_{ij}^{m+2} - k\theta_{ij}^{m+2}[L_h u_{ij}^{m+2} + g_{ij}^{m+2}] = u_{ij}^{m+1} + k\theta_{ij}^{m+1}[L_h u_{ij}^{m+1} + g_{ij}^{m+1}] \]

\[ = 2u_{ij}^{m+1} - (u_{ij}^m + k\theta_{ij}^m[L_h u_{ij}^m + g_{ij}^m]). \]
The second form of the second equation is the analogue of the Varga-type splitting of the second step of the Peaceman–Rachford method. Notice that for \((m+1+j)\) an even integer, this equation simply reduces to
\[
U_{ij}^{m+2} = 2U_{ij}^{m+1} - U_{ij}^m
\]
which allows us to define the fast hopscotch algorithm for this problem as follows:

**Computational Algorithm**

The algorithm is defined as proceeding from time level \(t = mk\) forwards in time. The steps are as follows. Notice that step (1) will only be used if either \(m = m_0\) (starting point) or print-out has been given at \(t = mk\).

1. For interior mesh points, with \((m+i+j)\) an odd integer, overwrite the value of \(u_{ij}^m\) with
\[
u_{ij}^{m+1} = u_{ij}^m + k[L_h u_{ij}^m + g_{ij}^m]
\]
(for all \(i, j\) satisfying \((m+i+j)\) odd). The above step is the restart procedure.

2. If the values of \(u_{ij}^{m+1}\) are required for print-out, overwrite the values of \(u_{ij}^m\) for \((m+1+i+j)\) odd by the formula
\[
u_{ij}^{m+1} - kL_h u_{ij}^{m+1} = u_{ij}^m + k g_{ij}^{m+1}.
\]
Notice that this defines the values of \(u_{ij}^{m+1}\) with \((m+1+i+j)\) odd, explicitly if the operator \(L_h\) is an \(E\)-operator. After print-out \(m\) is incremented by 1 and we return to step (1). If the values of \(u_{ij}^{m+1}\) are not required to be printed, proceed to step (3).

3. For a fixed value of \((i, j)\) with \((m+1+i+j)\) odd, define \(w_{ij}^m\) by
\[
u_{ij}^{m+1} - kL_h u_{ij}^{m+1} = u_{ij}^m + k g_{ij}^{m+1}.
\]
Notice that the values of \(u_{ij}^{m+1}, u_{ij}^{m+2}\) required in this formula have already been constructed in (1). Overwrite the value \(u_{ij}^m\) with \(u_{ij}^{m+2}\) defined by
\[
u_{ij}^{m+2} = 2u_{ij}^{m+1} - u_{ij}^m
\]
for this fixed value of \((i, j)\) with \((m+1+i+j)\) odd.

Repeat this process for all allowable \((i, j)\) with \((m+1+i+j)\) odd. Then increment \(m\) by 1.

If the values of \(u_{ij}^{m+1}\) are required for print-out, then return to step (2), otherwise return to the start of step (3) and repeat.

The above algorithm does not take into account the redefinition of boundary data but this may be easily incorporated at the start of step (2) or (3). As may be seen the process is very efficient in practice and extremely easy to program. We return to this topic in a later section.

It is worth noting that the hopscotch algorithm may be likened to the staggered mesh schemes used in fluid dynamical calculations. The interested reader is referred to the book by Richtmyer & Morton (1967) for further material on such schemes.

A further interesting point arises if one returns to equations (2.4a) and eliminates entirely the occurrence of the variable \(u_{ij}^{m+1}\). One generates a scheme which may be seen to be a Dufort–Frankel method (Dufort & Frankel, 1953). However, it must be stressed that this local equivalence does not infer the global equivalence of hopscotch and Dufort–Frankel schemes. The Dufort–Frankel scheme calculates each node by precisely the same formula starting from two given levels of information. The hopscotch process generates two sets of solutions but the starting process is automatic and not
equivalent to the Dufort-Frankel approach. However, the hopscotch process will only produce two sets of uncoupled solutions if \( L \) is an \( E \)-operator.

In order to prove convergence in the parabolic and elliptic cases we consider the hopscotch process \( \text{globally} \) and regard it as a two-step procedure. It is convenient to introduce several vectors and matrices at this stage.

Let \( u_{2m} \) denote the vector with components \( u_{ij}^{2m} \) for some ordering of internal points in \( \Omega \). (For \textit{more general} boundary conditions, it may be necessary to also include points in \( \partial \Omega \).) The normal ordering will be the usual one of ordering along rows for each fixed column. Let the matrix \( H \) be defined by

\[
[H u_{2m}]_{(i,j)} = -L u_{ij}^{2m}
\]
a sparse matrix in general.

We now define two diagonal matrices (of the same order as \( H \)) whose entries are either +1 or 0. Let \( I_1 \) be the diagonal matrix whose \((i,i)\) element is +1 if the \(i\)th entry in the vector \( u_{2m} \) is a mesh function \( u_{ij}^{2m} \) with \((r+s)\) an odd integer, and whose entries otherwise are zero. Let \( I_2 = I - I_1 \). Notice that for the normal ordering the matrix \( I_1 \) is composed of diagonal blocks whose entries are diagonal matrices with alternate ones and zeros. We note that \( I_1 I_2 = I_2 I_1 = 0 \), the null matrix and that \( I_1, I_2 \) satisfy

\[
H = I_1 H + I_2 H.
\]

Using this notation, we may define the two step hopscotch process (2.4a) globally in the form

\[
[I+kI_2 H]u_{2m+1} = [I-kI_1 H]u_{2m} + k(l_2 g_{2m+1} + l_1 g_{2m})
\]

\[
[I+kI_1 H]u_{2m+2} = [I-kI_2 H]u_{2m+1} + k(l_1 g_{2m+2} + l_2 g_{2m+1}).
\]

(2.5)

It is readily seen that this is an A.D.I process of the Peaceman-Rachford type (Varga, 1962) corresponding to the splitting of the matrix \( H \) into the form

\[
H = I_1 H + I_2 H.
\]

This formulation allows us to apply the known results on A.D.I. methods and, hopefully (since the splitting is of a particularly easy form) we may be able to obtain more powerful results concerning A.D.I. parameters.

An important point to notice is that the hopscotch process is \textit{always} of the form (2.5) irrespective of the number of space dimensions involved. This property therefore allows us to consider multidimensional problems (i.e. any number of space dimensions \( \geq 1 \)) in one theory. The conditions for convergence will \textit{solely} be governed by the coefficient matrix \( H \). In the following sections we carry out this analysis for (2.5) when applied to parabolic and elliptic problems. Similar procedures may be defined for first order hyperbolic systems. However, we will not consider such possibilities here but leave them to a future paper.

3. Linear Parabolic Problems

We now consider the solution of a general problem of the form (2.1) but initially restrict ourselves to those which give rise to a matrix \( H \) which is symmetric and positive definite.

The system (2.5) may be written in the form

\[
u_{2m+2} = T u_{2m} + b_{2m}
\]

(3.1)
where
\[ T = [I + kI_1 H^{-1}][I - kI_2 H][I + kI_2 H]^{-1}[I - kI_1 H] \]
\[ = [I + kI_1 H^{-1}][I - kI_2 H][I + kI_2 H]^{-1}[I - kI_1 H][I + kI_1 H][I + kI_1 H] \]
\[ = [I + kI_1 H^{-1}]T[I + kI_1 H]. \]

Also since $H$ is positive definite, it has a positive definite square root $H^*$ so that
\[ [I - kI_2 H][I + kI_2 H]^{-1} = H^{-1}[I - kH^*I_2 H^*][I + kH^*I_2 H^*]^{-1}H^* \]
\[ = H^{-1}H_2 H^*. \]

Therefore
\[ T = \{H^*[I + kI_1 H]\}^{-1}H_2 H_1 \{H^*[I + kI_1 H]\}. \]

Also the matrices $H^*I_1 H^*$ and $H^*I_2 H^*$ are non-negative definite since, for example,
\[ H^*I_1 H^* = (I^*H^*)^*H^*H^*. \]

We now state two lemmas of Kellogg (1964) which we employ later.

**Lemma 1**

If $\rho > 0$ and $(B + B^*)$ is non-negative definite then $\rho I + B$ has a bounded inverse and $\|(\rho I + B)^{-1}\| \leq \rho^{-1}$.

**Lemma 2**

If $\rho > 0$ and $(B + B^*)$ is non-negative definite, then the operator $(\rho I - B)(\rho I + B)^{-1}$ is bounded and has norm less than or equal to unity.

In order to use these two lemmas we must reconsider the form of our matrix $H$. Since this arises from a discretization of an elliptic partial differential operator of order 2, it follows that $H$ will be of the form $1/(h^2)$ where $\lim_{k \to 0} \hat{H} = \hat{H}$ a constant matrix. Let $r = k/h^2$ be the mesh ratio assumed constant. In what follows we will assume that $H$ has been replaced by $(1/h^2)\hat{H}$ but will maintain the notation $H$ for $\hat{H}$. Thus
\[ T = \{H^*[I + rI_1 H]\}^{-1}H_2 H_1 \{H^*[I + rI_1 H]\}. \]

where
\[ H_i = [I - rH^*I_1 H^*][I + rH^*I_1 H^*]^{-1} \]
for $i = 1, 2$. By lemma 2 it follows that for $r > 0$, constant, $\|H_i\| \leq 1$, $i = 1, 2$.

Now for stability we require that powers of $T$ be uniformly bounded. We have, by a standard argument, that
\[ T^m = \{H^*[I + rI_1 H]\}^{-1}(H_2 H_1)^m \{H^*[I + rI_1 H]\} \]

implying that
\[ \|T^m\| \leq \|(H^*[I + rI_1 H]\)^{-1}\| \|H^*[I + rI_1 H]\| \leq C \]
a constant independent of $m$.

Therefore we may state:

**Theorem 1**

The hopscotch process is stable for the solution of equation (2.1) if the matrix $H$ is positive definite.
We may generalize this theorem considerably by noticing that the powers of $T$ are bounded (using Kellogg's lemmas) if certain conditions are applied to the matrices $I_1H$ and $I_2H$. This gives

**Theorem 2**

The hopscotch process is stable for the solution of equation (2.1) if the matrices $I_1H$ and $I_2H$ both satisfy the condition

$$(u, Qu) + (Qu, u) \geq 0 \quad (Q = I_1H \text{ or } I_2H)$$

for all real $u \neq 0$ of appropriate order.

In order to cast the above theorem in a more workable form we require a further lemma:

**Lemma 3**

If the $n \times n$ non-singular matrix $H$ has $n$ independent left eigenvectors, then

(i) the matrices $I_1H$ and $I_2H$ also have $n$ independent left eigenvectors, and
(ii) the null spaces of $I_1H$ and $I_2H$ are disjoint.

**Proof**

Let $V_n$ be a vector space of $n$ dimensional real vectors. The matrix $I_1$ is a diagonal matrix with $p$ non-zero diagonal entries (equal to unity) and $I_2$ is a diagonal matrix with $(n-p)$ non-zero diagonal entries equal to unity. Thus, corresponding to the zero diagonal entries of $I_1$ there exist $(n-p)$ vectors of the set $\{e_j\}, j = 1, \ldots, n$, where $e_j$ is a vector whose only non-zero entry is a 1 in the $j$th position, which belong to the null space of the left eigenvectors of $I_1H$. Thus

$$d(N(I_1H)) \geq n-p$$

where $N(I_1H)$ is the null space of $I_1H$, and $d(U)$ is the dimension of $U$.

Similarly

$$d(N(I_2H)) \geq p.$$

Now since $I_1I_2 = I_2I_1 = 0$ we have that for any two $n$-dimensional vectors $u, v \in V_n$,

$$(I_1Hu, I_2Hv) = 0 \quad u, v \in V_n.$$

It follows that the two subspaces $V^{(1)}, V^{(2)}$ where

$$V^{(i)} = \{v: v = I_iHu, u \in V_n\}$$

are disjoint.

Also we have that

$$(*) \quad d(V^{(1)}) \leq p, \quad d(V^{(2)}) \leq n-p.$$

However since

$$d(V^{(1)} + V^{(2)}) + d(V^{(1)} \cap V^{(2)}) = d(V^{(1)}) + d(V^{(2)})$$

(a well known result) and since

$$d(V^{(1)} \cap V^{(2)}) = 0$$

we have that

$$d(V^{(1)}) + d(V^{(2)}) = n.$$

Therefore from $(*)$ it follows that

$$(**) \quad d(V^{(1)}) = p, \quad d(V^{(2)}) = n-p$$

$$d(N(I_1H)) = n-p, \quad d(N(I_2H)) = p,$$

and moreover we see that

$$N(I_1H) \cap N(I_2H) = \{0\}.$$
It follows immediately from (**) that the matrices $I_1H$ and $I_2H$ each have $n$ independent left eigenvectors.

Using this lemma, it is straightforward to show that a more workable form of theorem 2 is contained in

**Theorem 3**

If the non-singular matrix $H$ has a full set of independent eigenvectors, then the hopscotch process is stable if the eigenvalues of the two matrices $I_1H$, $I_2H$ have non-negative real parts.

It may easily be shown that it is *not* sufficient merely for the eigenvalues of the matrix $H$ to have non-negative real parts to guarantee that the eigenvalues of the matrices $I_1H$, $I_2H$ have non-negative real parts. (A simple counter example is the two by two matrix

$$
\begin{bmatrix}
    2 & 5 \\
    -1 & -1
\end{bmatrix}
$$

However, since the matrix $(-H)$ arises from the replacement of an elliptic operator, it has a certain structure attached to it. In particular, the matrix $H$ may have the property,

$$
(*) \quad h_{ii} \geq \sum_{j=1 \atop j \neq i}^{n} |h_{ij}| \quad x = 1, \ldots, n.
$$

where usually

$$
h_{ij} \leq 0 \quad i \neq j
$$

The above properties, in the case when $H$ has no eigenvector deficiency and is non-singular, are sufficient to allow us to prove

**Theorem 4**

The hopscotch process is stable if $H$ is non-singular, has a full set of eigenvectors and satisfies the conditions (*) above.

**Proof**

We need only show that the matrices $I_1H$, $I_2H$ have eigenvalues with non-negative real parts. Since these matrices are simply composed of rows from $H$ and null rows, it follows by applying Gershgorin’s theorem to each matrix $I_1H$, $I_2H$ and using conditions (*) that the conditions of theorem 3 are satisfied.

Notice that conditions (*) are simply diagonal dominance conditions frequently encountered in the basic replacement of an elliptic operator (see Bramble & Hubbard, 1963, for example).

It is a relatively easy matter to test conditions (*) above and, assuming that the matrix $H$ has no eigenvector deficiency, we can see that we have stability for a wide class of parabolic problems on general regions subject to general linear boundary conditions. If the matrix $H$ is a function of the space variables, then the analysis carries through immediately.

Having now demonstrated the stability of the hopscotch process, we prove its convergence by a standard argument.

We return to equation (2.4) and, defining

$$
e_{ij}^n = u_{ij}^n - U(ih, jh, mk)
$$
we have that

\[ [1 - k \theta_{ij}^{n+1} L_h] e_{ij}^{n+1} = [1 + k \theta_{ij}^n L_h] e_{ij}^n + d_{ij}^n, \]

where \( d_{ij}^n \) is the local truncation error. For a general mesh point it may easily be seen that

\[ d_{ij}^n = O(k^2 + kh^\sigma) \]

where \( \sigma \geq 1 \) and \( k/h^2 \) is constant (assuming \( U \) is sufficiently differentiable). In what follows, we will not attempt to obtain best estimates of convergence rate but simply indicate the convergence of the hopscotch process.

Using the notation above, it follows that

\[ e_{2m} = T e_{2m-2} + f_{2m-2} \]

where it may easily be shown that \( \| f_{2m-2} \| = O(k^2 + kh^\sigma) \) where \( \| \cdot \| \) denotes the usual \( L_2 \) norm. It follows that

\[ e_{2m} = T^m e_0 + \{ f_{2m-2} + T f_{2m-4} + \ldots + T^{m-1} f_0 \}. \]

Thus if \( t_{2m} = 2mk \) and the hopscotch process is stable, we have that

\[ \| e_{2m} \| \leq c \| e_0 \| + m C \max_i \| f_{2i} \| \]

so that

\[ \| e_{2m} \| \leq C \| e_0 \| + C' t_{2m} \cdot 0(k + h^\sigma) \]

for the worst possible (i.e. smallest) \( \sigma \).

By a similar argument, it follows that

\[ \| e_{2m+1} \| \leq C_1 \| e_0 \| + C_2 t_{2m+1} \cdot 0(k + h^\sigma). \]

Therefore, we may state

**Theorem 5**

If the hopscotch process is stable and if the local truncation error is no worse than \( O(k^2 + kh^\sigma) \) then the process has a convergence rate \( O(k + h^\sigma) \) as \( k, h \to 0 \) provided the appropriate mesh ratio \( k/h^2 \) is constant.

We note that we assumed in the above analysis that the appropriate mesh ratio \( k/h^2 \) was held constant. This is important. It is easily demonstrated computationally that the hopscotch process will not converge for the heat equation as \( k, h \to 0 \) if for example \( k/h \) is held constant. (See also Gordon, 1965.)

### 4. Linear Elliptic Problems

We turn now to a consideration of the convergence of the hopscotch process when used to determine iteratively the solution of a linear elliptic problem

\[ LU = g. \]

Using the normal replacements, this gives rise to the matrix system

\[ Hu = b \]

(4.1)

where we assume \( H \) is non-singular.

We define our hopscotch process in the form

\[ [wI + I_2 H] u_{2m+1} = [wI - I_2 H] u_{2m} + b \]

\[ [wI + I_1 H] u_{2m+2} = [wI - I_2 H] u_{2m+1} + b \]

(4.2)

which is, of course, precisely of the \( P - R \) form, and where \( w \propto 1/r \).
However, in this case we know more about the splitting of the matrix H in the form

$$H = I_1H + I_2H.$$ 

We now indicate how some general convergence theorems may be obtained.

In order that the above process (4.2) converge, we require that the spectral radius of $T$ be less than unity, where

$$T = [wI + I_1H][wI - I_2H][wI + I_2H]^{-1}[wI - I_1H]. \quad (4.3)$$

In fact, we may consider the more general procedure

$$[wD + I_2H]u_{2m+1} = [wD - I_1H]u_{2m} + b$$

$$[wD + I_1H]u_{2m+2} = [wD - I_2H]u_{2m+1} + b \quad (4.3)$$

where $D$ is a positive definite diagonal matrix. This time the iteration matrix is

$$T = [wD + I_1H]^{-1}[wD - I_2H][wD + I_2H]^{-1}[wD - I_1H].$$

If the matrix $H$ is positive definite, then the above matrix is similar to

$$\tilde{T} = [wI + H_1]^{-1}[wI - H_2][wI + H_2]^{-1}[wI - H_1] \quad (4.4)$$

where

$$H_1 = H^D - I_1HD - H^{-1}$$

$$H_2 = H^D - I_2 HD - H^{-1}$$

where it is easily seen that $H_2$ is similar to $(HD)^1I_1(HD)$ which is non-negative definite. Further, it is easy to verify that $N(H_1 + H_2) = \phi$ so we may apply the analysis of Kellogg & Spanier (1965) and Douglas & Pearcy (1963) to demonstrate the convergence of (4.3) and obtain "approximate" parameters. If $\alpha$ and $\beta$ are respectively lower and upper bounds on the smallest positive and largest eigenvalues of the matrices $H_1$ and $H_2$ then we can choose our parameters according to the well known rules, ignoring the fact that, in general, $H_1H_2 \neq H_2H_1$. In the hopscotch process, it is almost certain that a choice $D \neq I$ will be required to produce rapid convergence.

However, our assumption that $H$ has no eigenvector deficiency and the fact that $H$ and all vectors in the computation are real enables us to prove a more general convergence theorem which is a weakened form of theorem 1 of Douglas & Pearcy (1965). We first require an extension of Lemma 2.

**Lemma 6**

If $\rho > 0$ and $(B + B^*)$ is non-negative definite and $B$ has no eigenvector deficiency, then the operator $(\rho I - B)(\rho I + B)^{-1}$ satisfies

$$\|(\rho I - B)(\rho I + B)^{-1}u\| = \|u\|$$

if and only if $u \in N(B)$.

**Proof**

Consider

$$\frac{\|T(B)u\|^2}{\|u\|^2} = \frac{(T(B)u, T(B)u)}{(u, u)} = \left\langle ((\rho I - B)v, (\rho I - B)v), ((\rho I + B)v, (\rho I + B)v) \right\rangle$$

where $v = (\rho I + B)^{-1}u$.

By direct computation it follows that

$$\|T(B)u\|^2 = 1 \iff (Bv, v) + (v, Bv) = 0.$$
Now, since $B$ has no eigenvector deficiency, $(B + B^*)$ is non-negative definite and all quantities are real, a simple argument involving spectral decomposition of $v$ will show that $v \in N(B)$. But

$$u = (\rho I + B)v$$

hence $u \in N(B)$ as required.

Using Lemma 6 we obtain a weakened form of Lemma 2.3 of Douglas & Pearcy (1963).

**Lemma 7**

Let $[\alpha, \beta]$ be any interval such that $[\alpha, \beta] \subset (0, \infty)$ and suppose $(A + A^*)$ and $(B + B^*)$ are non-negative definite matrices such that $N(A) \cap N(B) = 0$. Suppose also that $A$ and $B$ have no eigenvector deficiency. Then there exists a positive number $\gamma < 1$ such that for any $a, b \in [\alpha, \beta]$ with $a \leq b$,

$$\|(A - aI)(A + bI)^{-1}(B - bI)(B + bI)^{-1}\| \leq \gamma.$$ 

The proof follows that given in Douglas & Pearcy (1963) using Lemma 6 to obtain the weaker result. Using these Lemmas we are able to state the following:

**Theorem 6**

Let $(I_1H + (I_1H)^*)$ and $(I_2H + (I_2H)^*)$ be non-negative definite matrices where $H$ has no eigenvector deficiency and let $D$ be a positive definite diagonal matrix of the same order as $H$. Let $\alpha < \beta$ be any positive real numbers. Then there exists a positive integer $t_0$ such that any sequence $\{w_k\}$ of the form

$$\beta = w_1 \geq w_2 \geq \ldots \geq w_{t_0} = \alpha$$

$$w_k = w_k \text{ (mod } t_0) \quad k > t_0$$

when used as the parameter sequence for the hopscotch process (4.3) will ensure that the sequence $\{u_{2m}\}$ will converge to a solution $u$ of (4.1).

**Proof**

The proof is a simple extension of the corresponding theorem in Douglas & Pearcy (1963). We merely mention the differences. We use Lemma 3 to guarantee that $D^{-1}I_1HD^{-1}, D^{-1}I_2HD^{-1}$, have no eigenvector deficiency. Then an application of Lemma 6 and the correspondingly extended Lemma 7 enables the theorem to be proved.

We notice that our requirements on the matrices $I_1H$ and $I_2H$ are precisely the same as those required for stability in the parabolic case. We are therefore able to guarantee convergence if $H$ satisfies the conditions of Theorem 4.

More refined estimates of convergence rate may be obtained by continuing the above analysis in the spirit of Douglas & Pearcy (1963) (see also Gary, 1967). We, however, leave the discussion at this point and consider how the hopscotch process for solving elliptic equations might be modified to improve convergence rates. Its main advantage lies in its extreme efficiency in storage and computational effort, as pointed out in Section 2. Even though its convergence rate may be slow in practice (though we have no strong reason to believe this need to be case), we can apply any of the devices outlined in the book of Wachspress (1966). These may result in an efficient inner–outer iteration based on the hopscotch process. For difficult problems, it will be interesting to see how this type of algorithm compares with the promising one of Stone (1968).
The hopscotch process enjoys yet another type of advantage when we consider non-linear problems. In this paper, we will content ourselves with a demonstration of ways of applying the hopscotch algorithm to such problems and show how the computational effort is again very small and that the process has certain nice properties.

In this section, we consider the hopscotch process applied to a non-linear $E$-operator $L_h(u)$. For definiteness, consider the cases of

(i) mild non-linearity when

$$L(u) = \nabla^2 u + g(x, y, u),$$  \hspace{1cm} (5.1)  

(ii) strong non-linearity

$$L(u) = \nabla (\alpha(u) \nabla u) + g(x, y, u).$$  \hspace{1cm} (5.2)

Only step (3) of the computational algorithm of Section 2 is affected by non-linearity. For the case of a mildly non-linear operator (e.g. (5.1)), this step requires the pointwise solution of an equation of the form

$$\alpha u + g(x, y, u) = \beta$$

where $\alpha, \beta$ are known. This is a particularly simple task compared with the solution of a set of coupled non-linear equations.

A straightforward Newton iteration would obviously provide rapid convergence. As a starting guess one could take the average of the nearest neighbours to the point $u$, as they are already known. On occasion such iteration may be unnecessary. For example, the well-known problem $\nabla^2 u = u^2$ when tackled by hopscotch gives rise to a pointwise quadratic equation in $u$. We note that, in fact, the implicitness only occurs at roughly half the points in the grid since step (3) is only used when $(m+1+i+j)$ is odd.

An alternative process which is also applicable to the strongly non-linear problems is to use the technique outlined above but without iteration. In other words, instead of solving, for example

$$\frac{1}{h^2} (\delta_x^2 + \delta_y^2) u_{ij} = (u_{ij})^2$$

solve

$$\frac{1}{h^2} (\delta_x^2 + \delta_y^2) u_{ij} = \left\{ u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i-1,j} \right\}^2$$

which is **explicit**. This technique of **averaging** is also suggested for non-linear problems such as (5.2). Notice that both these techniques enable one to keep storage to a minimum. It is proposed to test the two strategies above on a series of problems.

So far we have only been concerned with $E$-difference operators which arise from second order elliptic and parabolic equations. It is also possible to apply the hopscotch technique to non-linear hyperbolic first order systems but we leave such considerations to a future paper.

It is worth noting that Scala & Gordon (1966, 1968) have applied the original hopscotch type algorithm to fluid dynamical calculations with considerable success. We hope that the efficiency of the algorithm particularly in its present form will lead to its application to the numerical solution of many physical models. Its performance in the examples in Section 7 seems to suggest that it works very well in practice.
6. General Considerations

First of all let us consider the merits of the hopscotch algorithm over, for example, an A.D.I. process (Peaceman–Rachford). We summarize the advantages of the new algorithm when applied to the above problems:

(i) It has minimal storage requirements for problems of this type, since it overwrites core storage.

(ii) It is of the order of three to four times as fast, computationally, as the P-R process (a simple operations count demonstrates this fact). This is mainly due to the absence of tridiagonal inversions but is helped to a great extent by the efficiency of the algorithm in 2.

(iii) It is extremely easy to program both for linear and non-linear problems over general regions. It is almost possible to construct a general p.d.e. solver based on this algorithm!

(iv) It handles problems in many space dimensions almost as easily as those in one space dimension (of course, the incorporation of boundary conditions may not be as straightforward).

The main disadvantage is that in the elliptic case we have not as yet an optimum way of deriving parameters. This disadvantage exists also in the Peaceman–Rachford method but, apparently, not in Stone’s (1968) method.

So far no mention has been made of the replacement of boundary conditions. In general this causes little complication. In fact, if we use centred approximations, the process is very easy. Thus, for example, to replace

\[ \frac{\partial u}{\partial x} + bu = g(x) \]

use

\[ \frac{a}{2h}[u_{i+1} - u_{i-1}] + \frac{1}{2}b[u_{i+1} + u_{i-1}] = g_i \]

(It is not necessary that such centering be used, merely easier in application.)

The grid used throughout this paper was the normal rectangular grid. It is just as easy to determine a hopscotch process if one uses, for example, a hexagonal grid. If one uses a triangular grid, then the algorithm becomes implicit in that certain tridiagonal systems must be solved.

If the finite difference scheme is not an \( E \)-operator, then a certain degree of implicitness is introduced. For certain operators, the degree of implicitness may be considerable (such as for the nine point replacement of Laplace) or slight (only tri-diagonal inversion is required along diagonals for mixed derivative problems for certain replacements). Also, even for \( E \)-difference operators, it is possible to define the matrices \( I_1 \) and \( I_2 \) differently so that the explicit character of the scheme is not maintained. For example, a row-implicit scheme (involving tri-diagonal systems) is obtained if we define \( I_1 \) to have a zero in its \( l \)th position, if the \( l \)th entry in the vector \( u_n \) has an even value of \( j \). It is not yet obvious what the best choice of \( I_1, I_2 \) is, though the definition earlier in the paper would appear to be most efficient.

Also, it is not necessary to use \( H = I_1H + I_2H \). Equally, one might have

\[ H = HI_1 + HI_2 \]
though, in the case of the parabolic equation, this produces certain unusual schemes. Thus for \( \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \) we have

\[
[1 - r \delta_t^2 \partial_t^{m+1}] u_{t+1}^m = [1 + r \delta_t^2 \partial_t^m] u_t^m
\]

which gives the schemes

\[
\begin{align*}
 u_{t+1}^m - u_t^m &= r(u_{t+1}^{m+1} + u_{t-1}^{m+1} - 2u_t^m) \\
 u_t^{m+1} - u_t^m &= r(u_{t+1}^m + u_{t-1}^m - 2u_t^{m+1})
\end{align*}
\]

which are respectively conditionally stable \( 0 < r \leq \frac{1}{2} \) and unconditionally stable \( r > 0 \). Again, it is not clear whether such an approach offers any advantage.

If the operator \( L \) is time dependent, then the process is well-defined if we simply replace the quantity

\[
\theta_j L u_{ij}, \quad s = m, m+1
\]

in (2.4a) (and equivalently in (2.5)) by the quantity

\[
\theta_j L u_{ij}, \quad s = m, m+1.
\]

7. Some Numerical Examples

To demonstrate the above procedures and to provide a comparison with existing methods, several experiments were undertaken. In the main, these consisted of solving a Dirichlet problem for the inhomogeneous heat equation

\[
\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + f(x, y, t)
\]

(7.1)

over various regions. The first set consisted of solving (7.1) where

\[
f(x, y, t) = \sin x \sin y \, e^{-t} - 4
\]

and

\[
u(x, y, t) = \sin x \sin y \, e^{-t} + x^2 + y^2
\]

over various shaped regions. Four algorithms were used, namely the hopscotch process, the Peaceman–Rachford A.D.I. method, the locally one-dimensional method "equivalent" to the P.R. method (see Gourlay & Mitchell, 1969) but without the necessary transformations, and the backwards L.O.D. method as outlined in Hubbard (1965). (No boundary corrections were employed.)

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hopscotch</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>(a)</td>
</tr>
<tr>
<td>(b)</td>
</tr>
<tr>
<td>(c)</td>
</tr>
<tr>
<td>(d)</td>
</tr>
<tr>
<td>(e) (i)</td>
</tr>
<tr>
<td>(e) (ii)</td>
</tr>
</tbody>
</table>

The four regions considered were: (a) a right angled triangle; (b) a diamond; (c) a circle; (d) an ellipse.

In the case of the regions (c) and (d) suitable variants of the five point Laplacian were used near the boundary but the boundary was in no way deformed.
The average errors in the respective runs are quoted in Table 1. (The distribution of errors was the same for all methods.)

The runs were undertaken with a grid (where uniform) of size \( h = 0.1 \) and 100 steps were computed with \( r = 1.0 \) so that \( T = 1.0 \). The hopscotch process was roughly three times faster than any of the others, even though the programs were perhaps not as efficient as they might have been. The programming of hopscotch was an order of magnitude easier than the programming of the other three methods.

The same programs were run for an L-shaped region with

\[
f(x, y, t) = 0
\]

This leads to

\[
u(x, y, t) = e^{-8\pi^2 t} \sin 2\pi x \sin 2\pi y + r \sin \frac{3\pi}{2} \theta
\]

where the polar co-ordinates \((r, \theta)\) have their origin at the re-entrant corner. (The \((x-y)\) origin was at the other corner on this diagonal.) This problem has singularities in the first and higher derivatives with respect to \( r \). The errors at points (i) close to the re-entrant corner, (ii) away from the re-entrant corner, are quoted in Table 1 under (e) (i) and (e) (ii) respectively (again after 100 steps with above grid sizes).

From Table 1 it can be easily seen that the hopscotch process and the Peaceman–Rachford method are of the same order of accuracy. The comparatively poor results for the locally one dimensional method and the backward difference L.O.D. method, demonstrate that these methods must be used with great care in practice. Further, both these L.O.D. methods require roughly the same amount of computer time as does the Peaceman–Rachford method. Of course, their accuracy could be improved by employing the procedures in Gourlay & Mitchell (1969) though this would increase the running time considerably. In practice, it would appear that, for parabolic problems, one must make the choice between A.D.I. and hopscotch. The ease of programming, economy in storage and efficiency in machine time weigh heavily in favour of the hopscotch process.

We now consider an elliptic problem. In fact, we choose to solve the two point boundary value problem

\[
y'' = f(x, y, y')
\]

with suitable conditions at, say, \( x = a, b \). Using the averaging process outlined in the previous section, it follows that we may use

\[
L_{\delta}(y_i) = \frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}) - f(x_i, \frac{1}{2}(y_{i+1} + y_{i-1}), \frac{1}{2h}(y_{i+1} + y_{i-1}))
\]

which is an \( E \)-operator.

As an illustrative problem, we consider the “difficult” example of Keller (1968),

\[
(x^2 y')' = \frac{x^2 y}{\varepsilon(y + \kappa)}, \quad y'(0) = 0, \quad y(1) = 1.
\]

For this example the replacement is

\[
\frac{1}{h^2} \delta_{x^2}(x^2 \delta_{x} y_i) = \frac{x_i^2}{\varepsilon} \frac{(y_{i+1} + y_{i-1})}{(y_{i+1} + y_{i-1} + 2\kappa)}
\]

together with \( y_{-1} = y_1, \quad y_N = 1 \), \((N = 100)\) and the choice \( \kappa = \varepsilon = 0.1 \), is made.

It was found by experiment that, when the fastest convergence occurred, just over
400 iterations of hopscotch were required (constant single parameter). This required 210 seconds of running time on the University of Dundee NCR-ELLIOTT 4130 machine. This convergence is obviously not fast but the speed of the process makes up for this deficiency (which can probably be reduced by determining a suitable sequence of parameters). Our only aim in carrying out this example was to verify that convergence would occur in a reasonable time for a difficult non-linear problem.

To show that convergence may be considerably faster, the same program was used to solve the model problem,

\[ y'' = 0 \quad y'(0) = 0 \quad y(1) = 1 \]

with \( h = \frac{1}{10} \). The fastest convergence this time occurred in only 43 iterations of the hopscotch process (again with constant parameter). Our results tend to show that the convergence rate is proportional to \( N \), the number of grid points, and that the optimum parameter is proportional to \( N \) also. A similar type of algorithm (but not equivalent) has been noted in Pope (1960) for the solution of two point boundary problems.

8. Conclusions
A fast algorithm has been proposed and studied. It is particularly suitable for evolutionary problems and is very efficient both man and machine-wise. It appears to work also for difficult elliptic problems though more investigation is required into parameter choice. Its main characteristics are its efficiency in machine-time, its economy of storage, its simplicity in programming and its ability to handle non-linear problems in a straightforward manner.

REFERENCES