JOURNAL OF COMPUTATIONAL PHYSICS 15, 188–199 (1974)

The Efficient Computation of the Heat Distribution in a 5 × 5 Matrix Thermal Print Head

JOHN LL. MORRIS AND IAN F. NICOLL

Department of Mathematics, University of Dundee, Dundee, Scotland

Received July 17, 1973

The odd–even hopscotch algorithm is considered for the solution of a heat equation with constant coefficients representing the heat flow in an isotropic thermal print head comprising a 5×5 matrix of elements. The results of experiments on the simulation of alphabetic character printing are reported.

1. INTRODUCTION

One of the recent advances in printing technology has been the development of the thermal printer. The printing is performed by the direct application of heat onto thermosensitive paper in which a chemical reaction takes place at temperatures above a certain threshold. The heat is radiated from a thermal print head composed of a matrix of heat elements each of which consists of a heat resistor embedded in a thin film of a good conducting material surmounting a glass substrate. The thin film allows the rapid diffusion of heat over the printing surface when the heat resistor is switched on and when switched off the glass substrate acts as a heat sink. Thus by switching on and off specific elements a series of characters can be produced on the paper.

This printing technique can be made faster than conventional mechanical devices *if* an optimal "time on/time off" cycle for the switching of the heat elements in the thermal print head can be found. In addition the physical properties of the materials used in the manufacture of the thermal print head need to be considered carefully as, for a given set of physical parameters, a given on–off switching of the heat sources can cause an overall rise in the print head temperature which in turn results in indistinct characters being produced on the print paper.

A mathematical model for the thermal print head problem was first proposed by Chen [1]. Such was the complex nature of the model that a general analytical solution was impossible and thus a numerical solution was required. In the paper by Chen [1] the substrate problem was solved using an explicit finite difference method and the thin film problem using an implicit scheme. In a more recent paper [4], both Alternating Directional Implicit (A.D.I.) and Locally One

Dimensional (L.O.D.) methods were used. Experiments were carried out in [1] and [4] to test the accuracy of the numerical methods for the thermal print head problem with a known theoretical solution and also to investigate the action of the print head under more realistic physical conditions. However, the implicit schemes require large computer storage and hence restrict the size of problem under consideration to that of a single element in the print matrix and thus make the simulation of character printing impractical.

In a more recent paper [5] the authors considered a class of hopscotch methods (see Gourlay [2] and also Saul'yev [6]) for the solution of the heat flow in a single element of an isotropic print head and it was concluded that, although there was little difference in accuracy between the schemes, the odd–even hopscotch algorithm had a computational superiority and required minimal computer storage.

In this paper we shall report on the simulation of character printing, using the odd–even hopscotch scheme to solve the heat flow problem in a thermal print head comprising a 5×5 matrix of elements. In Section 2 we shall briefly describe the mathematical model of Chen and the odd–even hopscotch method. The results of the experiments, carried out on a 5×5 matrix of heat elements to illustrate the crucial effect of the choice of the physical parameters, are reported in Section 3. The paper is concluded in Section 4.

2. The Mathematical Model and the Numerical Method of Solution

The mathematical model of the thermal print head described in [1] is unusual in two respects. First, the thin film is assumed so thin and to have appropriate conductivity properties so that no temperature gradient exists in the z-direction. This has the effect of producing a thin film which is assumed to have no dimension in the z-direction other than to give the thin film a thermal capacity due to its physical thickness D. Second, the solution of the heat flow problem in the thin film constitutes a boundary condition for the solution of the total print head.

The region in which the solution is required is defined by

$$\bar{R} = R \times [0 < t \leqslant T],$$

where $R = \{(x, y, z): 0 < x, y < l, 0 < z < b\}$ with $l = Mb$ the overall length of the side of an $M \times M$ print head matrix with each element of side b. We denote the boundary of $\bar{R}$ by $\partial\bar{R}$.

The equation governing the temperature distribution $u = u(x, y, z, t)$ in the glass substrate $(0 < z \leqslant b)$ is

$$\frac{\partial u}{\partial t} = \frac{\kappa_1}{\rho_1 C_1}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \tag{1}$$

subject to the initial condition

$$u(x, y, z, 0) = 0.0, \tag{2}$$

and the boundary conditions

$$u(x, y, b, t) = 0.0, \tag{3}$$

$$\frac{\partial u}{\partial \eta} = 0 \quad \text{on} \quad \partial \bar{R}_{x=0,l} \quad \text{and} \quad \partial \bar{R}_{y=0,l}, \tag{4}$$

and the values $u = u(x, y, 0, t)$, where u is the solution of the heat equation for the thin film

$$\frac{\partial u}{\partial t} = \frac{\kappa}{\rho C}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{\delta(x, y, t)}{\rho C} - \frac{h_0}{D\rho C}(u - u_\infty) \tag{5}$$

subject to the initial/boundary conditions (2) and (4) evaluated at $z = 0$. η is the outward drawn normal to the edges of the thermal print head and the physical parameters κ, ρ, C and κ_1, ρ_1, C_1 denote the thermal conductivity, density, and specific heat of the thin film and substrate, respectively. u_∞ is the ambient temperature and h_0 is the convective heat transfer coefficient between the thin film and air.

The multiple switching of the heat sources in the $M \times M$ matrix of elements is defined by the term

$$\begin{aligned}\delta(x, y, t) = q &\left\{1 - \sum_{\nu=0}^{2n-2} (-1)\, H(t - t_\nu)\right\} \\ &\times \left\{\sum_{\lambda=0}^{M-1} \sum_{\mu=0}^{M-1} [H(x - \lambda b - a) - H(x - (\lambda + 1)b + a)] \right. \\ &\left. \times [H(y - \mu b - a) - H(y - (\mu + 1)b + a)]\right\},\end{aligned}$$

where $H(\theta)$ is the Heaviside function defined by

$$H(\theta) = \begin{cases} 0 & \theta < 0, \\ 1 & \theta > 0, \end{cases}$$

n is the number of on/off switchings and the t_ν are defined such that $\{t_\nu: \nu = 0, 2, 4, \ldots, 2n - 2\}$ are the switch-off times and $\{t_\nu: \nu = 1, 3, 5, \ldots, 2n - 3\}$ are the switch-on times for the heat sources. The heat resistors, each outputting q watts per unit surface area, are defined by the squares $\{(x, y): \lambda b + a \leqslant x \leqslant (\lambda + 1)b - a, \mu b + a \leqslant y \leqslant (\mu + 1)b - a, 0 \leqslant \lambda, \mu \leqslant M - 1\}$ where a is the distance between the edge of the print elements and the edge of the enclosed resistor. For simplicity we shall assume the heat resistors cover the whole area of each print element, that is $a = 0$.

To obtain a numerical solution to Eqs. (1) and (5) with their associated initial/boundary conditions we first superimpose a rectilinear grid on the region of computation $\bar{R}$ where the spacings in the space variables are taken equal, namely $\Delta_x = \Delta_y = \Delta_z = h$, and the mesh spacing in the time dimension Δ_t is denoted by τ. The mesh ratio $r = \tau/h^2$ is taken to be constant throughout. We denote by u_{ijk}^m the value of the unknown u at the point $(ih, jh, kh, \tau m) = (x, y, z, t)$, $i, j = 0, 1, \ldots, MN, k = 0, 1, \ldots, N, m = 0, 1, 2, \ldots,$ where $MNh = l$ and $Nh = b$.

Writing Eq. (5) in the form

$$\frac{\partial u}{\partial t} = L_1 u + d(x, y, t),$$

where $L_1 = (u/\rho C)((\partial^2/\partial x^2) + (\partial^2/\partial y^2)) - (h_0/D\rho C)$ is a linear elliptic differential operator and $d(x, y, t) = (\delta(x, y, t)/\rho C) + (h_0 u_\infty/D\rho C)$, the associated odd–even hopscotch method may be written as

$$u_{ij0}^{m+1} = u_{ij0}^m + \tau L_h u_{ij0}^m + \tau d_{ij}^m \qquad \text{for} \quad i + j + m \text{ even}, \tag{6}$$

$$u_{ij0}^{m+1} - \tau L_h u_{ij0}^{m+1} = u_{ij0}^m + \tau d_{ij}^{m+1} \qquad \text{for} \quad i + j + m \text{ odd}, \tag{7}$$

or

$$u_{ij0}^{m+1} - \tau\theta_{ij}^{m+1} L_h u_{ij0}^{m+1} = u_{ij0}^m + \tau\theta_{ij}^m L_h u_{ij0}^m + \tau(\theta_{ij}^{m+1} d_{ij}^{m+1} + \theta_{ij}^m d_{ij}^m), \tag{8}$$

where

$$\theta_{ij}^m = \begin{cases} 1 & \text{for} \quad i + j + m \text{ even}, \\ 0 & \text{for} \quad i + j + m \text{ odd}, \end{cases}$$

where L_h is a "suitable" (see below) difference approximation to L_1 of Eq. (5).

In a similar manner, writing Eq. (1) in the form

$$\frac{\partial u}{\partial t} = L_2 u,$$

where L_2 is the three-dimensional elliptic operator $L_2 = (\kappa_1/\rho_1 C_1)((\partial^2/\partial x^2) + (\partial^2/\partial y^2) + (\partial^2/\partial z^2))$, the three-dimensional odd–even hopscotch algorithm is given by

$$u_{ijk}^{m+1} = u_{ijk}^m + \tau L_h u_{ijk}^m \qquad \text{for} \quad i + j + k + m \text{ even}, \tag{9}$$

$$u_{ijk}^{m+1} - \tau L_h u_{ijk}^{m+1} = u_{ijk}^m \qquad \text{for} \quad i + j + k + m \text{ odd}, \tag{10}$$

or

$$u_{ijk}^{m+1} - \tau\theta_{ijk}^{m+1} L_h u_{ijk}^{m+1} = u_{ijk}^{m} + \tau\theta_{ijk}^{m} L_h u_{ijk}^{m}, \tag{11}$$

where

$$\theta_{ijk}^{m} = \begin{cases} 1 & \text{for } i+j+k+m \text{ even,} \\ 0 & \text{for } i+j+k+m \text{ odd,} \end{cases}$$

and again L_h is a "suitable" difference approximation to L_2 of Eq. (1).

The two- and three-dimensional odd–even hopscotch algorithms proceed as described in [2]. Namely, for the alternate nodal points at which $\theta_{ijk}^{m} = 1$ the numerical solution at the advanced time step u_{ijk}^{m+1} is calculated by means of the explicit schemes (6) and (9). The numerical solution at the remaining nodal points given by $\theta_{ijk}^{m} = 0$ is now calculated by the implicit schemes (7) and (10). However, if the difference operators in Eqs. (7) and (10) are members of a class of E operators, as defined by Gourlay in Ref. [2], then it is found that the solutions at those points about u_{ijk}^{m+1}, which usually make the scheme implicit, have already been computed using Eqs. (6) and (9) so that the schemes (7) and (10) are now computationally explicit.

If we define the finite difference operator L_h in Eq. (8) to be the E operator $L_h = (\kappa/\rho C)((\delta_x{}^2/h^2) + (\delta_y{}^2/h^2)) - (h_0/D\rho C) \equiv L_1 + O(h^2)$ and similarly the operator L_h in the three-dimensional scheme (11) to be the E operator $L_h = (\kappa_1/\rho_1 C_1)((\delta_x{}^2/h^2) + (\delta_y{}^2/h^2) + (\delta_z{}^2/h^2)) \equiv L_2 + O(h^2)$, where δ_x, δ_y, δ_z are the usual central difference operators defined by

$$\delta_x u_{ijk}^{m} = u_{(i+\frac{1}{2})jk}^{m} - u_{(i-\frac{1}{2})jk}^{m}, \qquad \text{etc.}$$

then the computational algorithm for the numerical solution of the thermal print head may be seen more clearly if we write out explicitly equations (9) and (10).

For Eq. (9) we have

$$u_{ijk}^{m+1} = u_{ijk}^{m} + \frac{\tau}{h^2}\frac{\kappa_1}{\rho_1 C_1} \times [u_{i+1jk} + u_{i-1jk} + u_{ij+1k} + u_{ij-1k} + u_{ijk+1} + u_{ijk-1} - 6u_{ijk}]^m. \tag{9.a}$$

For Eq. (10) we have

$$u_{ijk}^{m+1} - \frac{\tau}{h^2}\frac{\kappa_1}{\rho_1 C_1}[u_{i+1jk} + u_{i-1jk} + u_{ij+1k} + u_{ij-1k} + u_{ijk+1} + u_{ijk-1} - 6u_{ijk}]^{m+1} = u_{ijk}^{m}. \tag{10.a}$$

Now, Eq. (9.a) is applied for all those points for which $i + j + k + m$ is even. Hence for all the points at $t = (m + 1)\tau$ which have a sum of subscripts and superscript which is odd, the difference solution is known.

Equation (10.a) is then applied for $i + j + k + m$ odd. Thus the values in the square brackets of Eq. (10.a) have a sum of subscripts and superscripts which is odd, i.e., these are the very point values first calculated by Eq. (9.a). Hence the scheme is computationally explicit.

We have described briefly this part of the computational method (the full details may be found in Ref. [2]) to stress the fact that the odd–even hopscotch algorithm allows one to dispense with the inversion of large tridiagonal matrices which arise when, for example, numerical methods such as A.D.I. or L.O.D. schemes are employed. Thus the hopscotch method requires correspondingly small storage requirements for computation.

For the E operators described above the hopscotch method is unconditionally stable and has a local accuracy of $O(h^2 + \tau)$. By virtue of the method, no intermediate boundary conditions are required, and consequently no boundary correction techniques, as described in Ref. [3], are required.

Finally we apply the normal boundary conditions (4) by using the simple difference replacements

$$\left.\frac{\partial u}{\partial x}\right|_{x=0} = \frac{u_{1jk}^{m} - u_{-1jk}^{m}}{2h}$$

and

$$\left.\frac{\partial u}{\partial x}\right|_{x=l} = \frac{u_{(N+1)jk}^{m} - u_{(N-1)jk}^{m}}{2h}$$

with similar expressions for $(\partial u/\partial y)|_{y=0}$ and $(\partial u/\partial y)|_{y=l}$.

3. The 5×5 Element Print Head

Several numerical experiments using the odd–even hopscotch algorithm were carried out on an isotropic print head comprising a 5×5 matrix of heat elements to simulate the printing of physical characters. As previously mentioned in Section 1, care must be taken in the choice of the physical parameters for the print head as, for a given on/off switching, there is a possibility of an overall temperature rise in the printing surface which would cause a smudging effect. To illustrate this, the heat elements were switched on and off in a pattern which produced the alphabetic letters H, I, X in turn, using a print cycle of eight time steps—6 on, 2 off.

For the parameter set, $a = 0$, $b = 1$, $l = 5$, $h = 1/6$, $r = 1.0$, $\kappa_1 = \kappa_2 = 0.5$, $\rho = 24.42$, $C = 2.6$, $\kappa_3 = \kappa_4 = \kappa_5 = 0.05$, $\rho_1 = 0.13$, $C_1 = 3.0095$, $q = 10.0$,

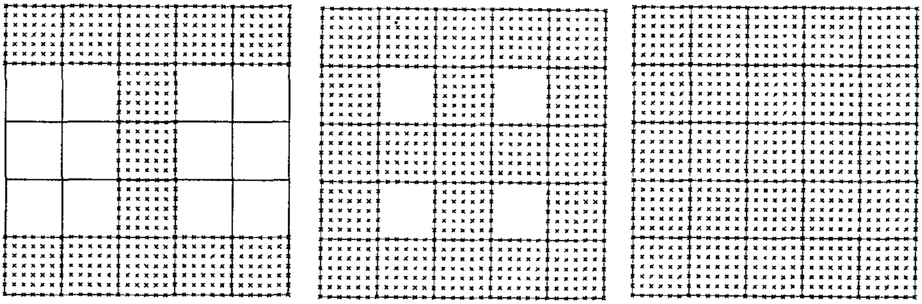

FIG. 1. (a) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 6\tau$ with appropriate elements to produce letter H being switched on for $0 \leqslant t \leqslant 6\tau$. (b) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 14\tau$ with appropriate elements to produce letters H and I being switched on for $0 \leqslant t \leqslant 6\tau$ and $8\tau \leqslant t \leqslant 14\tau$, respectively. (c) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 22\tau$ with appropriate elements to produce letters H, I, X being switched on for $0 \leqslant t \leqslant 6\tau$, $8\tau \leqslant t \leqslant 14\tau$, $16\tau \leqslant t \leqslant 22\tau$, respectively.

$h_0 = 0.000679$, $u_\infty = 0.0$, $D = 0.00011045$, Figs. 1 show that temperature pattern on the printing surface at times 6τ, 14τ, 22τ, respectively, where the symbol $\times$ denotes the points where the temperature u exceeds the threshold temperature. With these parameters, at time $t = 14\tau$ the letters H and I coalesce and also at time 22τ the letters H, I, X are seen to coalesce.

Let (p, q), $p, q = 1, 2, 3, 4, 5$ denote the row and column position of a heat element in the print head. Figure 2 shows the temperature distribution at the points $x = (p - \frac{1}{2})b$, $y = (q - \frac{1}{2})b$ on the surface of each heat element (p, q), $p, q = 1, 2, 3$ which, due to the symmetry of the characters H, I, X about the lines $x = 2.5b$ and $y = 2.5b$, gives a complete picture of the temperature distribution at the central points of all the elements.

The graph for the element (2, 1) which is switched on in the first print cycle for the letter H and off thereafter shows that there is a negligible heat loss after the switch off time indicating that the smudging was brought about by an impractical choice of physical constants.

A "good" parameter set is $a = 0$, $b = 1$, $l = 5$, $h = 1/6$, $r = 1.0$, $\kappa_1 = \kappa_2 = 1.0$, $\rho = 10.49$, $C = 0.0556$, $\kappa_3 = \kappa_4 = \kappa_5 = 0.0028$, $\rho_1 = 2.4$, $C_1 = 0.2$, $q = 10.0$, $h_0 = 0.000053$, $u_\infty = 0.0$, $D = 0.000015$ which with the same print cycle gives clearly defined characters Fig. 3. The points at which the temperature is greater than the threshold are denoted by X.

The temperature distributions for the heat elements (p, q) $p, q = 1, 2, 3$ Fig. 4 show that when a heat element is switched off the temperature rapidly falls beneath the threshold.

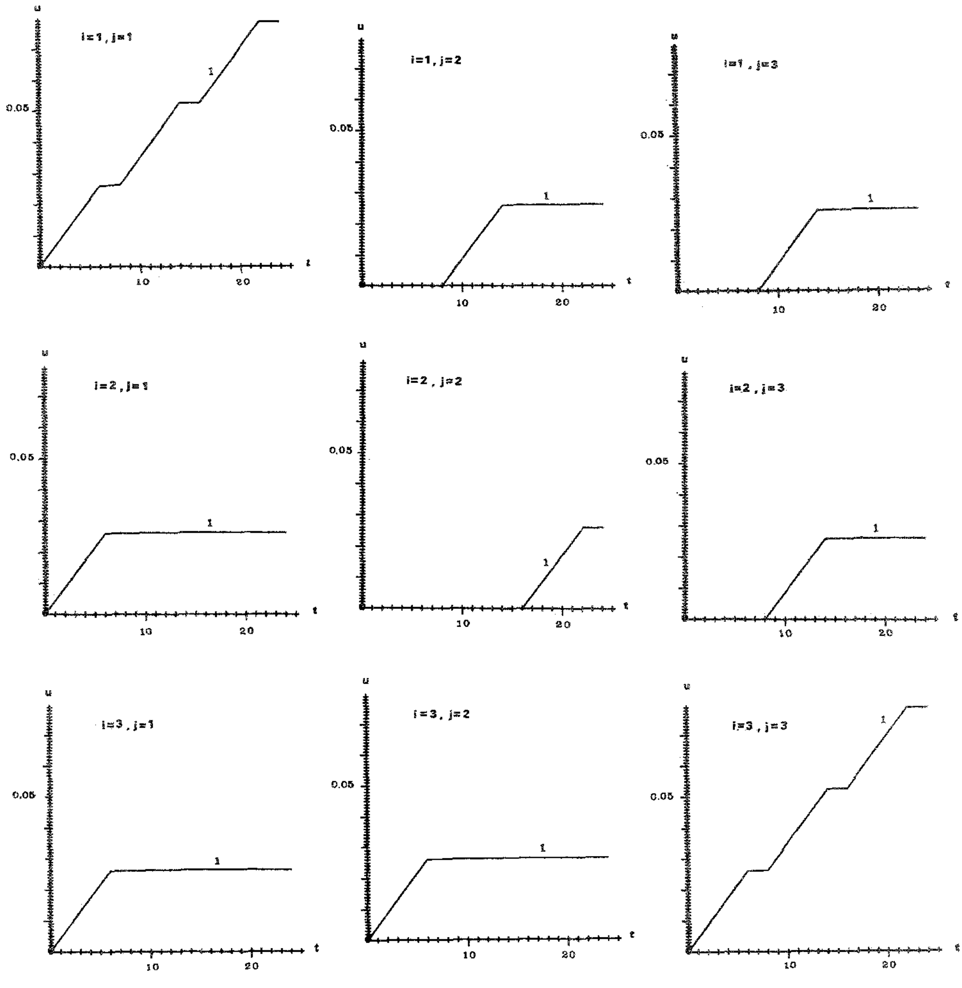

FIG. 2. Graphs of the temperature distributions on the print surface at the central points of the heat elements (i, j), $i, j = 1, 2, 3$ for a print cycle of 8τ to produce the characters H, I, X with smudging.

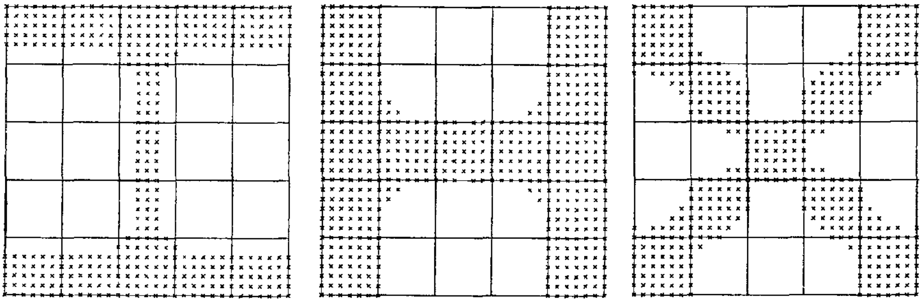

FIG. 3. (a) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 6\tau$ with appropriate elements to produce letter H being switched on for $0 \leqslant t \leqslant 6\tau$. (b) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 14\tau$ with appropriate elements to produce letters H and I being switched on for $0 \leqslant t \leqslant 6\tau$ and $8\tau \leqslant t \leqslant 14\tau$ respectively. (c) Points on the surface of a thermal print head where the temperature exceeds the threshold at time $t = 22\tau$ with appropriate elements to produce letters H, I, X being switched on for $0 \leqslant t \leqslant 6\tau$, $8\tau \leqslant t \leqslant 14\tau$, $16\tau \leqslant t \leqslant 22\tau$, respectively.

4. Concluding Remarks

The results of the previous section report only on two of a series of experiments carried out by the authors to observe the effect of the variation of the physical parameters on the simulation of character printing. These results show clearly that the correct choice of physical parameter values is crucial to ensure the production of distinct characters by the thermal printer.

The computational experiments reported in Section 3 and in [4] were carried out at the University of Dundee on an Elliott 4130 computer with a $64K$ word store.

In [4], the thermal print head problem was computed using A.D.I. and L.O.D. methods. It is the conclusion of the present authors that the hopscotch method described here represents a considerable saving in programming effort. In addition the hopscotch method represents a considerable saving in computing, both in time and in computer storage. The exact ratios of computational efficiency of the respective methods is difficult to ascertain precisely. The programs were all written in ALGOL 60 so that considerable dynamic storage allocation is used and no actual storage usage is ever given by the operating system. However we found that with the remaining store from the $64.K$ word memory not taken up by the operating system, we were unable to compute any significantly larger model than the single $(10 \times 10 \times 10)$ point element described in [4]. By the hopscotch method the 5×5 element thermal print head was computed using $20K$ mesh points. This 20:1 ratio in storage is not precise since by careful programming the storage utilization of the A.D.I. and L.O.D. methods could be reduced, somewhat, at a greatly increased cost in computation. An approximate comparison of the storage

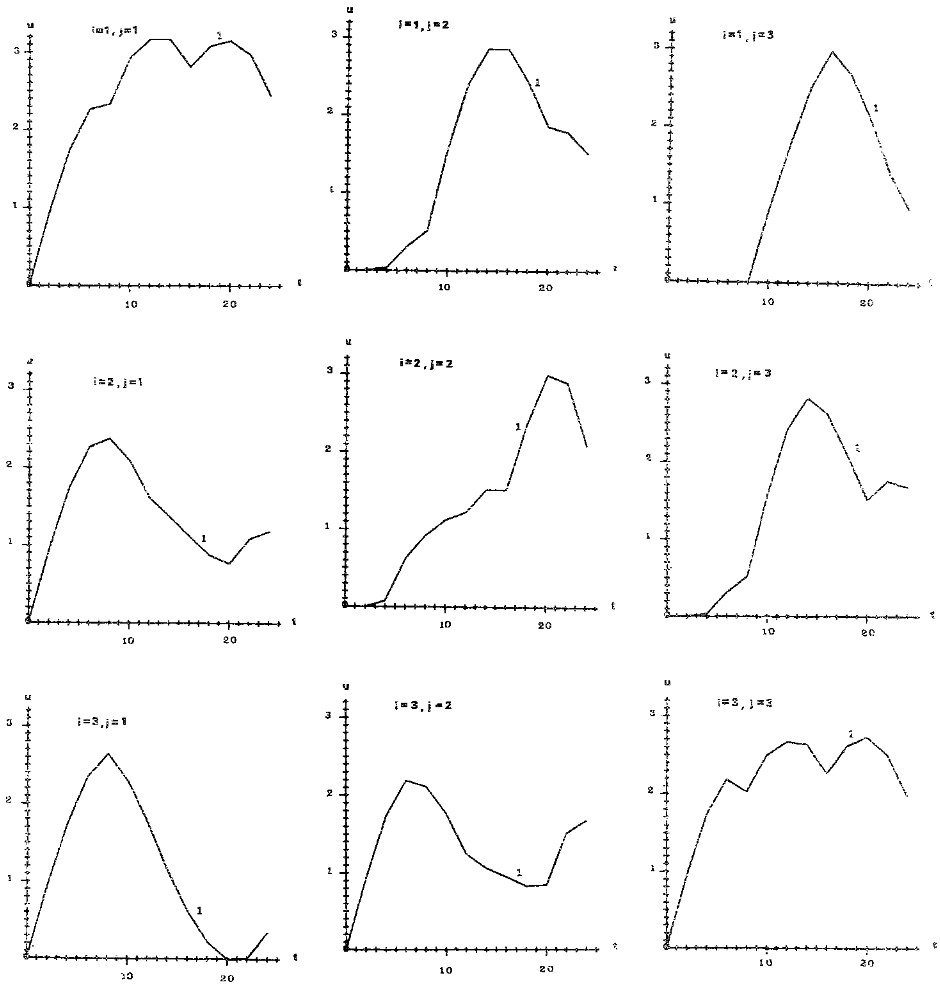

FIG. 4. Graphs of the temperature distributions on the print surface at the central points of the heat elements (i, j) $i, j = 1, 2, 3$ for a print cycle of 8τ to produce the character H, I, X.

allocations can be found as follows. The hopscotch method as given by Eqs. (9) and (10) requires only a single vector of length 1000 for the single heating element. Since there is no matrix solution this is essentially all of the storage. In addition, the code to calculate the solution is extremely short and the logic simple. In comparison the A.D.I. and L.O.D methods (in the notation of [4]) are splittings of the system of equations $(I + rU)(I + rV)(I + rW)\,\mathbf{v}_{m+1} = (I - rU)(I - rV)(I - rW)\,\mathbf{v}_m + \mathbf{k}$, where $\mathbf{k}$ is a vector of the same dimension as $\mathbf{v}$ the solution, i.e., (1×1000), and contains the contributions from boundaries. The matrix $(I + rU)$, etc. are 1000×1000 matrices (for the 1000 point model). Clearly these are not stored. However each matrix has three nonzero bands which are stored in three (1×1000) vectors. In all we have 12 vectors of length 1000. Each element of a *real* vector constitutes two words of storage. When pointers are taken into account the actual storage per (1×1000) vector is in excess of $2K$ *words*. In fact for the L.O.D. this is increased to 13 by virtue of the transformation methods necessary to implement the L.O.D. method in a manner to ensure $O(\tau^2)$ accuracy. In addition to these vectors additional vectors are needed in the calculation of the tridiagonal systems as described in Varga [7], although not being as large as the 12 vectors, still contribute to the storage allocation. Since $O(\tau^2)$ accuracy is being sought it was necessary to implement a predictor corrector version of the implciit methods used in the thin film—this contributed an extra vector of length 100. Hence, approximately, on storage of vectors alone 15 times as much storage was necessary for the splitting methods as was needed by the hopscotch scheme. The A.D.I., and particularly the L.O.D., codes were extremely long in comparison with the hopscotch code and required of the order of $10K$ words. When the operating system is added it is clear that the computer store is just about filled with the single element for L.O.D. and A.D.I. We admit that less storage and more repeated calculation would have allowed a bigger model but this increased computer cost was not warranted—the hopscotch algorithm seemed to be the correct approach.

We would like to stress the fact that it would not be feasible to obtain the results of Section 3 by using either A.D.I. or L.O.D. methods as, for a realistic computer storage size, we would be restricted to too few points in the 5×5 matrix of elements to provide data of any practical value.

We conclude therefore that the odd–even hopscotch algorithm is a method which is particularly useful for complicated physical problems in which the associated coefficients are isotropic. However we point out the possible dangers in using the method for *anisotropic* problems (see [5]). In such cases the line or A.D.I. variants of the hopscotch method would be advocated with their associated increase in accuracy, but with an increased storage requirement. These storage requirements, however, are still minimal in comparison with the needs of A.D.I. and L.O.D. methods and the hopscotch methods can hence still be used to solve the full matrix print head problem described here.

Acknowledgments

Mr Nicoll's share of the work was carried out while he was in receipt of a Dundee University research studentship.

References

1. L. K. Chen, Unsteady, three-dimensional, conductive heat transfer analysis of thin film composites, N.C.R. (Dayton) Report, June 1967.
2. A. R. Gourlay, A fast second order partial differential equation solver, *J. Inst. Math. Appl.* **6** (1970).
3. A. R. Gourlay and A. R. Mitchell, Intermediate boundary correction for split operator methods in three dimensions, *B.I.T.* **7** (1967).
4. J. Ll. Morris, On the numerical solution of a heat equation associated with a thermal print head, *J. Comp. Phys.* **5** (1970).
5. J. Ll. Morris and I. F. Nicoll, Hopscotch methods for an anisotropic thermal print head problem, *J. Comp. Phys.* **13** (1973).
6. V. K. Saul'yev, Integration of equations of the parabolic type by the method of nets, *Pergamosh* (1964).
7. R. S. Varga, "Matrix Iterative Analysis," Prentice-Hall, Englewood Cliffs, NJ, 1962.