Journal of Computational Physics 228 (2009) 2175–2193

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

# An efficient cell-centered diffusion scheme for quadrilateral grids

M.M. Basko [a,*,1], J. Maruhn [b], An. Tauschwitz [b]

[a] GSI, Darmstadt, Germany
[b] Frankfurt University, Frankfurt, Germany

ARTICLE INFO

*Article history:*
Received 5 June 2008
Received in revised form 6 November 2008
Accepted 26 November 2008
Available online 9 December 2008

*Keywords:*
Diffusion equation
Quadrilateral grid
Cell-centered scheme

ABSTRACT

A new algorithm for solution of diffusion equations in two dimensions on structured quadrilateral grids is proposed. The algorithm is based on a semi-implicit method for the time discretization and has a nine-point local stencil in space. Our scheme is fast, quite accurate and demonstrates good spatial convergence. The presented numerical tests show that it is well suited for hydrocodes with cell-centered principal variables.

## 1. Introduction

In this paper we describe a numerical algorithm for solution of the diffusion equation in two dimensions on quadrilateral grids, which appears to provide an excellent compromise between simplicity of realization, efficiency (in terms of the required computer time) and accuracy. Efficiency becomes a key issue if the diffusion solver is to be incorporated into a two-dimensional (2D) or three-dimensional (3D) hydrodynamics code.

From general considerations it is clear that major improvements in efficiency for multi-dimensional schemes can be achieved if one chooses (i) a semi-implicit rather than fully implicit approach with respect to time differencing and (ii) a linear spatial differencing scheme. As had been proven by Kershaw [1], a linear second-order scheme on an arbitrary 2D mesh cannot be monotone. Monotonicity can be restored with non-linear algorithms [2,3], but such algorithms entail iterative solution of a large system of non-linear equations and are rather costly. Aiming at high performance efficiency, we stay by linear approach and compare our algorithm with the best earlier published schemes from this class [4–6]. Departures from monotonicity do not appear to be a serious issue in practical applications.

Ideally, one would prefer to use a first-order fully implicit time discretization, which is robust, unconditionally stable and sets no additional time-step limit. However, solution of the corresponding large system of linear equations becomes a real challenge for certain versions of spatial discretization [5], and is usually rather costly in two and three dimensions. Here, a major simplification can be achieved by employing a symmetric semi-implicit (SSI) method proposed by Livne and Glasner [7], which is easy to implement and quite efficient in terms of megaflops per time step. But the decisive advantage of this method emerges when one tries to incorporate even the simplest version of the radiation transport equation into a multi-dimensional hydrodynamics code with thermal conduction [8]. From the hydrodynamics point of view, radiation transport in the quasi-steady approximation is just an additional mechanism of non-local heat transport. As a consequence, the matrix

[*] Corresponding author.
*E-mail address:* basko@itep.ru (M.M. Basko).
*URL:* http://www.basko.net (M.M. Basko).
[1] On leave from ITEP, 117218 Moscow, Russian Federation.

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.031

of the corresponding large linear (or non-linear) system for new temperatures under a fully implicit treatment of both the thermal conduction and radiation transport is no longer sparse, which renders this approach almost impractical. This latter consideration served as a principal motivation for basing our diffusion solver on the SSI method.

Strictly speaking, the algorithm described in this paper cannot be claimed as a new numerical method because it is composed of known constituent parts. Also, it is not very general because it is restricted to structured quadrilateral (in two dimensions) grids. However, to the best of our knowledge, the algorithm as a whole has neither been published nor properly investigated before. Our results indicate that this algorithm has certain important advantages, which make it quite promising for application in multi-dimensional radiation hydrodynamics codes.

## 2. Basic equations

The present algorithm for solution of the diffusion equation is constructed with the primary aim to be implemented into a 2D hydrodynamical code based on a Godunov-type method, where the mass density $\rho$, the material velocity $\mathbf{u}$ and the mass-specific total (internal plus kinetic) energy $E = e + \frac{1}{2}u^2$ are the primary dependent variables. More specifically, our diffusion solver is aimed at description of the thermal heat conduction. The general form of the hydrodynamic energy equation with thermal diffusion is

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot [(\rho E + p)\mathbf{u}] = \nabla \cdot (\kappa \nabla T) + Q, \tag{1}$$

where $p = p(\rho, e)$ is the pressure, $T = T(\rho, e)$ is the temperature, $e$ is the mass-specific internal energy, $\kappa$ is the heat conduction coefficient, and $Q = Q(t, \mathbf{x})$ is the volume-specific heating rate due to external energy sources.

We assume that at each Lagrangian time step, where the total energy $E$ is advanced to a new time level $t + \Delta t$, the contributions to the energy increment from the $p\,dV$ work and from thermal conduction are evaluated separately, by using the same "old" values of the principal variables from the previous time step, and then summed up. Thus, the conduction phase of the algorithm handles the energy redistribution among the Lagrangian cells according to the right-hand side of Eq. (1) (the external heating $Q$ is also treated at this phase). Because it is accomplished with fixed values of $\rho$ and $\mathbf{u}$, the conduction phase must provide a solution to the diffusion equation

$$\rho c_V \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) + Q \tag{2}$$

on a motionless grid – which may be rather distorted due to prior hydrodynamic steps. In Eq. (2) we actually switch over to a new primary dependent variable $T = T(t, \mathbf{x})$. Because in the Godunov-type methods the primary variables $\rho$, $\mathbf{u}$ and $E$ are naturally assigned to cell centers, we need a numerical algorithm for Eq. (2) based on cell-centered values of the temperature $T$. The mass-specific heat capacity $c_V$ at constant volume is assumed to be known and also cell-centered.

## 3. Grid notation

We solve Eq. (2) in two dimensions on a quadrilateral grid, which is logically rectangular. The grid is composed of a set of nodes (vertices of quadrilaterals) $\mathbf{x}_{ij} = (x_{ij}, y_{ij}),\ i = 1, 2, \ldots, n_x,\ j = 1, 2, \ldots, n_y$, in a global orthogonal coordinate system $(x, y)$. The numbering convention is such that the cell $(i,j)$ is a quadrilateral defined by the nodes $(i,j),\ (i+1,j),\ (i,j+1)$ and $(i+1,j+1)$; vertex $(i,j)$ is always the lower-left corner of cell $(i,j)$; see Fig. 1.

We reserve two possibilities for the metric of our two-dimensional $(x, y)$ space, namely, (i) the Cartesian metric of $dV = dx\,dy\,dz$, and (ii) the cylindrical $(r, z)$ metric of $dV = r dr\,dz\,d\phi$, where $V$ is the three-dimensional (3D) volume. In the

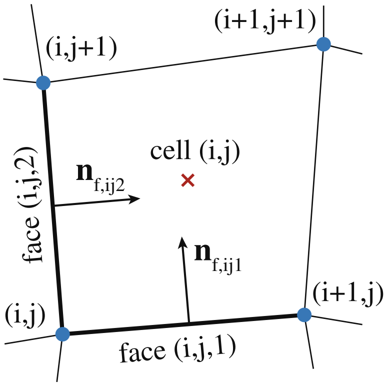

**Fig. 1.** Numbering of grid cells, cell vertices and faces.

cylindrical case we can set either $(x,y) \equiv (r,z)$ or $(x,y) \equiv (z,r)$. The three possibilities can be combined by introducing a global cylindrical radius

$$R = R(x,y) = \begin{cases} 1, & \text{Cartesian metric,} \\ x, & \text{cylindrical metric with } (x,y) \equiv (r,z), \\ y, & \text{cylindrical metric with } (x,y) \equiv (z,r), \end{cases} \tag{3}$$

and using the expression

$$V_{ij} = \int_{\text{cell}(i,j)} R dx dy \tag{4}$$

for the cell volume $V_{ij}$. Here, $V_{ij}$ is the 3D cell volume either per unit length along the $z$-axis in the Cartesian case, or per radian of the azimuth angle $\phi$ in the cylindrical case.

Each grid cell has four edges, which are called faces. Faces are numbered after vertices, i.e. we associate two faces $(i,j,m)$, $m = 1,2$, with each vertex $(i,j)$: face $(i,j,1)$ connects vertex $(i,j)$ with vertex $(i+1,j)$, face $(i,j,2)$ connects vertex $(i,j)$ with vertex $(i,j+1)$. Each face $(i,j,m)$ has a unit normal vector $\mathbf{n}_{f,ijm}$, which is perpendicular to face $(i,j,m)$ and points inside the cell $(i,j)$; see Fig. 1. The position of the geometric cell center is defined as

$$\mathbf{x}_{c,ij} = \frac{1}{4}\left(\mathbf{x}_{ij} + \mathbf{x}_{i+1,j} + \mathbf{x}_{i,j+1} + \mathbf{x}_{i+1,j+1}\right). \tag{5}$$

## 4. Numerical algorithm

### 4.1. Conservative SSI scheme

Let $T_{ij}$ be the cell-centered values of temperature on the above defined grid at time $t$, which we want to advance according to Eq. (2) to new values $\widetilde{T}_{ij}$ at time $t + \Delta t$. The key quantity that we use to discretize Eq. (2) is the integrated (i.e. in [erg s$^{-1}$]) energy flux $H_{ijm}$ across the face $(i,j,m)$, defined as

$$H_{ijm} = -\int_{\text{face}(i,j,m)} (\kappa \nabla T \cdot \mathbf{n}_{f,ijm}) R dl, \tag{6}$$

where $dl$ is the length element along face $(i,j,m)$. We assume that the old temperatures $T_{ij}$ and the old values $\kappa_{ij}$ (cell-centered) of the conduction coefficient are used to discretize Eq. (6) (spatial discretization; see the next paragraph). In this way we obtain the old (not advanced) fluxes $H_{ijm}$ that are linear functions of old temperatures $T_{ij}$. Then, an obvious (see Fig. 2) fully conservative explicit discretization of Eq. (2) will be

$$c_{V,ij} M_{ij} \tau_{ij} = \Delta t \left(H_{ij1} + H_{ij2} - H_{i,j+1,1} - H_{i+1,j,2} + q_{ij} M_{ij}\right), \tag{7}$$

where

$$\tau_{ij} = \widetilde{T}_{ij} - T_{ij}, \tag{8}$$

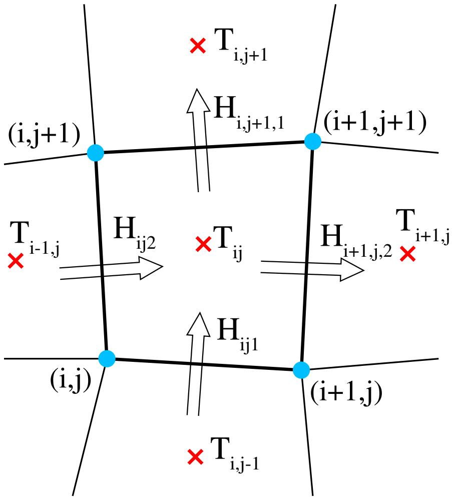

**Fig. 2.** Explicit (old) heat fluxes $H_{ijm}$ across the faces of cell $(i,j)$.

is the temperature increment in cell $(i,j)$,

$$M_{ij} = \int_{\text{cell}(i,j)} \rho R dx dy = \rho_{ij} V_{ij} \tag{9}$$

is the mass of this cell, and

$$q_{ij} = Q_{ij}/\rho_{ij} \tag{10}$$

is the mass-specific external heating rate.

As is well known, the fully explicit scheme (7) is unpractical because of severe stability constraints on the time step $\Delta t$ [9]. An alternative fully implicit scheme, where the new fluxes $\tilde{H}_{ijm}$, expressed in terms of the unknown advanced temperatures $\widetilde{T}_{ij}$, are used on the right-hand side of Eq. (7), and which is unconditionally stable for any $\Delta t$, becomes rather complex and computationally expensive in multiple dimensions because of necessity to solve a large system of coupled linear equations for $\widetilde{T}_{ij}$.

As a practical compromise, a symmetric semi-implicit (SSI) method was proposed in Ref. [7]. In this scheme the new fluxes $\widetilde{H}_{ijm}, \widetilde{H}_{i,j+1,1}, \widetilde{H}_{i+1,j,2}$, that are to be used on the right-hand side of Eq. (7), are calculated by using only one new temperature $\widetilde{T}_{ij}$ in the central cell $(i,j)$, and old temperatures in all the neighboring cells. However, one immediately realizes that in this approach one has to deal with two different fluxes associated with each face: one $\widetilde{H}^{+}_{ijm}$ calculated from above the face $(i,j,m)$, and the other $\widetilde{H}^{-}_{ijm}$ calculated from below the same face $(i,j,m)$. More specifically, we have to introduce the fluxes

$$\widetilde{H}^{+}_{ijm} = H_{ijm} - a_{ijm}\tau_{ij}, \tag{11}$$

$$\widetilde{H}^{-}_{ijm} = H_{ijm} + b_{ijm} \cdot \left\{ \begin{matrix} \tau_{i,j-1}, & m = 1, \\ \tau_{i-1,j}, & m = 2, \end{matrix} \right\}, \tag{12}$$

where

$$a_{ijm} = -\frac{\partial H_{ijm}}{\partial T_{ij}}, \tag{13}$$

$$b_{ij1} = \frac{\partial H_{ij1}}{\partial T_{i,j-1}}, \quad b_{ij2} = \frac{\partial H_{ij2}}{\partial T_{i-1,j}}. \tag{14}$$

The coefficient $a_{ijm}$ is the minus derivative of the explicit flux $H_{ijm}$ with respect to the forward (in the direction of $\mathbf{n}_{f,ijm}$) cell temperature, while $b_{ijm}$ is the derivative of the same explicit flux with respect to the corresponding backward cell temperature. By their physical meaning, both $a_{ijm}$ and $b_{ijm}$ must be non-negative. The authors of Ref. [7] present a mathematical proof that the SSI method is unconditionally stable.

Once we get two unequal fluxes across the same face, we loose energy conservation. Energy conservation can be restored if we calculate the amount of energy $\tilde{\delta}_{ij}$ (cell-associated) lost at a given time step, and redeposit it into the same cell at the next time step. As a result, we arrive at the following conservative SSI discretization scheme of Eq. (2)

$$c_{V,ij} M_{ij} \tau_{ij} = \Delta t \left( \widetilde{H}^{+}_{ij1} + \widetilde{H}^{+}_{ij2} - \widetilde{H}^{-}_{i,j+1,1} - \widetilde{H}^{-}_{i+1,j,2} + q_{ij} M_{ij} \right) + \delta_{ij}, \tag{15}$$

where $\delta_{ij}$ is the SSI energy correction from the previous time step. Eq. (15), combined with Eqs. (11) and (12), is easily solved with respect to $\tau_{ij}$,

$$\tau_{ij} = \frac{(H_{ij1} + H_{ij2} - H_{i,j+1,1} - H_{i+1,j,2} + q_{ij} M_{ij})\Delta t + \delta_{ij}}{c_{V,ij} M_{ij} + (a_{ij1} + a_{ij2} + b_{i,j+1,1} + b_{i+1,j,2})\Delta t}, \tag{16}$$

to yield the advanced temperatures $\widetilde{T}_{ij} = T_{ij} + \tau_{ij}$.

Once $\tau_{ij}$ are known, we can calculate the energy corrections $\tilde{\delta}_{ij}$ for the next time step. First, we find the energy $\tilde{\delta}_{ijm}$ lost at face $(i,j,m)$,

$$\tilde{\delta}_{ijm} = \left( H^{-}_{ijm} - H^{+}_{ijm} \right) \Delta t = \left[ a_{ijm} \tau_{ijm} + b_{ijm} \cdot \left\{ \begin{matrix} \tau_{i,j-1}, & m = 1, \\ \tau_{i-1,j}, & m = 2, \end{matrix} \right\} \right] \Delta t. \tag{17}$$

Then, we split this energy into two parts: a fraction $0 \leqslant \chi_{ijm} \leqslant 1$ goes into the forward-lying cell, and a fraction $1 - \chi_{ijm}$ goes into the backward-lying cell. As a result, the energy correction that must be redeposited in cell $(i,j)$ at the next time step is given by

$$\tilde{\delta}_{ij} = \chi_{ij1} \tilde{\delta}_{ij1} + \chi_{ij2} \tilde{\delta}_{ij2} + (1 - \chi_{i,j+1,1}) \tilde{\delta}_{i,j+1,1} + (1 - \chi_{i+1,j,2}) \tilde{\delta}_{i+1,j,2}. \tag{18}$$

The split weights $\chi_{ijm}$ are assumed to be proportional to the bulk heat capacities $C^{\pm}_{\Delta,ijm}$ of the two triangles lying on face $(i,j,m)$ and having their top vertices at the geometric centers of the forward-lying and backward-lying adjacent cells – as it is shown in Fig. 3. More specifically,

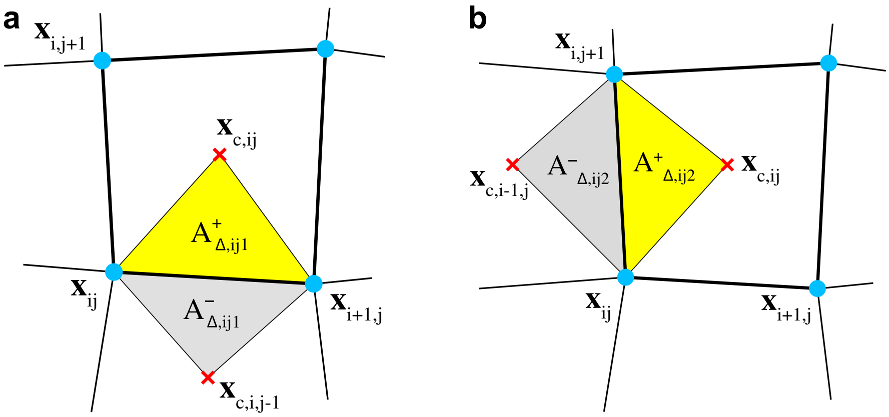

**Fig. 3.** Face-adjacent triangles used to calculate the weights for splitting the face-centered energy correction $\tilde{\delta}_{ijm}$, and the weights for the face-centered conduction coefficient $\kappa_{f,ijm}$.

$$\chi_{ijm} = \frac{C^+_{\Delta,ijm}}{C^+_{\Delta,ijm} + C^-_{\Delta,ijm}}, \tag{19}$$

where

$$C^+_{\Delta,ij1} = \frac{1}{3}\left(R_{ij} + R_{i+1,j} + R_{c,ij}\right)\, A^+_{\Delta,ij1}\; c_{V,ij}\; \rho_{ij}, \tag{20}$$

$$C^-_{\Delta,ij1} = \frac{1}{3}\left(R_{ij} + R_{i+1,j} + R_{c,i,j-1}\right)\, A^-_{\Delta,ij1}\; c_{V,i,j-1}\; \rho_{i,j-1}, \tag{21}$$

$$C^+_{\Delta,ij2} = \frac{1}{3}\left(R_{ij} + R_{i,j+1} + R_{c,ij}\right)\, A^+_{\Delta,ij2}\; c_{V,ij}\; \rho_{ij}, \tag{22}$$

$$C^-_{\Delta,ij2} = \frac{1}{3}\left(R_{ij} + R_{i,j+1} + R_{c,i-1,j}\right)\, A^-_{\Delta,ij2}\; c_{V,i-1,j}\; \rho_{i-1,j}, \tag{23}$$

$A^\pm_{\Delta,ijm}$ are the areas of the corresponding triangles adjacent to face $(ijm)$ (see Fig. 3), $R_{ij}$ and $R_{c,ij}$ are, respectively, the global cylindrical radii of the vertex $(i,j)$ and of the geometric center of cell $(i,j)$.

### 4.2. Face-centered fluxes

Now we turn to a finite-difference approximation of fluxes $H_{ijm}$ defined by Eq. (6). First, we denote

$$\mathbf{g} = \nabla T, \tag{24}$$

and introduce two relevant face-associated grid vectors

$$\mathbf{l}_{v,ijm} = \begin{cases} \mathbf{x}_{i+1,j} - \mathbf{x}_{i,j}, & m = 1, \\ \mathbf{x}_{i,j+1} - \mathbf{x}_{i,j}, & m = 2, \end{cases} \tag{25}$$

$$\mathbf{l}_{c,ijm} = \begin{cases} \mathbf{x}_{c,ij} - \mathbf{x}_{c,i,j-1}, & m = 1, \\ \mathbf{x}_{c,ij} - \mathbf{x}_{c,i-1,j}, & m = 2. \end{cases} \tag{26}$$

Vector $\mathbf{l}_{v,ijm}$ connects the two end vertices of face $(i,j,m)$, whereas vector $\mathbf{l}_{c,ijm}$ connects the centers of the two cells adjacent to face $(i,j,m)$; see Fig. 4. Then, an obvious second-order discretization of Eq. (6) will be

$$H_{ijm} = -\kappa_{f,ijm}\left(\mathbf{g}_{f,ijm}\cdot\mathbf{n}_{f,ijm}\right)|\mathbf{l}_{v,ijm}|R_{f,ijm}, \tag{27}$$

where

$$R_{f,ijm} = \begin{cases} \frac{1}{2}\left(R_{ij} + R_{i+1,j}\right), & m = 1, \\ \frac{1}{2}\left(R_{ij} + R_{i,j+1}\right), & m = 2, \end{cases} \tag{28}$$

and $\kappa_{f,ijm}$ and $\mathbf{g}_{f,ijm}$ are, respectively, the conduction coefficient and the temperature gradient evaluated at the midpoint of face $(i,j,m)$.

To derive a formula for $\mathbf{g}_{f,ijm}$, we assume for a moment that $\kappa$ is continuous across the face $(i,j,m)$, and that we know not only the cell-centered temperatures $T_{ij}$, but also the temperatures $T_{v,ij}$ at cell vertices. Then, the two components of vector $\mathbf{g}_{f,ijm}$ can be found from two rather obvious equations

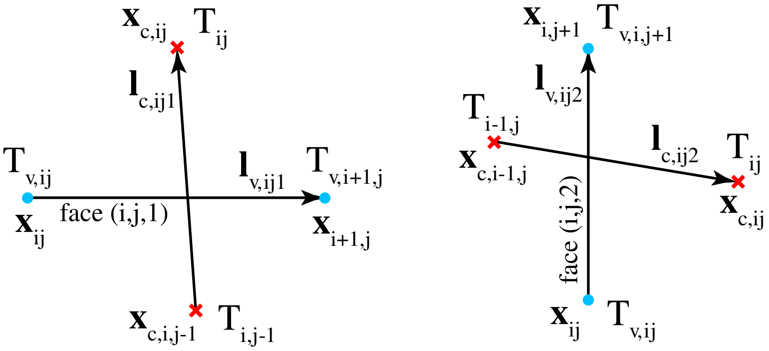

**Fig. 4.** Vector scheme for temperature gradient evaluation.

$$\begin{cases} \mathbf{g}_{f,ijm} \cdot \mathbf{l}_{v,ijm} = \Delta T_{v,ijm}, \\ \mathbf{g}_{f,ijm} \cdot \mathbf{l}_{c,ijm} = \Delta T_{c,ijm}, \end{cases} \tag{29}$$

where we have denoted

$$\Delta T_{v,ijm} = \begin{cases} T_{v,i+1,j} - T_{v,i,j}, & m = 1, \\ T_{v,i,j+1} - T_{v,i,j}, & m = 2, \end{cases} \tag{30}$$

$$\Delta T_{c,ijm} = \begin{cases} T_{ij} - T_{i,j-1}, & m = 1, \\ T_{ij} - T_{i-1,j}, & m = 2. \end{cases} \tag{31}$$

By solving the linear system (29) with respect to the two components of $\mathbf{g}_{f,ijm}$ and substituting the result into Eq. (27), we obtain the final expression for $H_{ijm}$,

$$H_{ijm} = \frac{\kappa_{f,ijm} R_{f,ijm}}{\left|\mathbf{l}_{v,ijm} \times \mathbf{l}_{c,ijm}\right|} \left[ \left(\mathbf{l}_{v,ijm} \cdot \mathbf{l}_{c,ijm}\right) \Delta T_{v,ijm} - \left|\mathbf{l}_{v,ijm}\right|^2 \Delta T_{c,ijm} \right]. \tag{32}$$

Now we can relax the assumption of a continuous conduction coefficient. With $\kappa$ being discontinuous across a face $(i,j,m)$, physics requires the normal component of the flux $-\kappa\mathbf{g}$ to be continuous. As a result, the normal component of the gradient $\mathbf{g}$ becomes discontinuous, and we have three unknown components of the vector $\mathbf{g}_{f,ijm}$. If we write

$$\mathbf{g}^{\pm}_{f,ijm} = g^{\pm}_{\perp} \mathbf{n}_{f,ijm} + g_{\parallel} \frac{\mathbf{l}_{v,ijm}}{|\mathbf{l}_{v,ijm}|}, \tag{33}$$

we obtain instead of Eq. (29) the following three equations

$$\begin{cases} \kappa^- g^-_{\perp} = \kappa^+ g^+_{\perp}, \\ g_{\parallel} |\mathbf{l}_{v,ijm}| = \Delta T_{v,ijm}, \\ 2g^-_{\perp} A^-_{\Delta,ijm} + 2g^+_{\perp} A^+_{\Delta,ijm} + g_{\parallel} \left(\mathbf{l}_{v,ijm} \cdot \mathbf{l}_{c,ijm}\right) = |\mathbf{l}_{v,ijm}| \Delta T_{c,ijm}, \end{cases} \tag{34}$$

to calculate the three unknown components $g^-_{\perp}, g^+_{\perp}, g_{\parallel}$ of the temperature gradient $\mathbf{g}_{f,ijm}$ at face $(i,j,m)$. Here, $\kappa^{\pm}$ are the two values of the conduction coefficient $\kappa$ on two sides of face $(i,j,m), A^{\pm}_{\Delta,ijm}$ are the areas of the two adjacent triangles shown in Fig. 3. Once we solve the system (34) and take into account that $2(A^-_{\Delta,ijm} + A^+_{\Delta,ijm}) = |\mathbf{l}_{v,ijm} \times \mathbf{l}_{c,ijm}|$, we obtain the same expression (32) for $H_{ijm}$, with $\kappa_{f,ijm}$ given by Eq. (43), i.e. by a weighted harmonic mean of the two cell-centered coefficients $\kappa_{ij}$ in the two adjacent cells.

Note that in the particular case of a rectangular mesh, when we have $\mathbf{l}_{v,ijm} \cdot \mathbf{l}_{c,ijm} = 0, |\mathbf{l}_{v,ijm} \times \mathbf{l}_{c,ijm}| = |\mathbf{l}_{v,ijm}| \cdot |\mathbf{l}_{c,ijm}|$, Eq. (32) becomes particularly simple, the values of the vertex temperatures $T_{v,ij}$ are not needed, and our algorithm reduces to the standard second-order central five-point scheme.

### 4.3. Vertex temperatures

To close the finite-difference approximation of fluxes $H_{ijm}$, we need an interpolation scheme for the vertex temperatures $T_{v,ij}$. In the logical space of grid indices $(i,j)$ each vertex $(i,j)$ is surrounded by four cell centers with physical coordinates $\mathbf{x}_{c,ij}$, $\mathbf{x}_{c,i-1,j}$, $\mathbf{x}_{c,i,j-1}$ and $\mathbf{x}_{c,i-1,j-1}$, which have known values of temperatures $T_{ij}$, $T_{i-1,j}, T_{i,j-1}$ and $T_{i-1,j-1}$; see Fig. 5. We want a linear interpolation formula of the form

$$T_{v,ij} = \mu_{1,ij} T_{ij} + \mu_{2,ij} T_{i-1,j} + \mu_{3,ij} T_{i-1,j-1} + \mu_{4,ij} T_{i,j-1}, \tag{35}$$

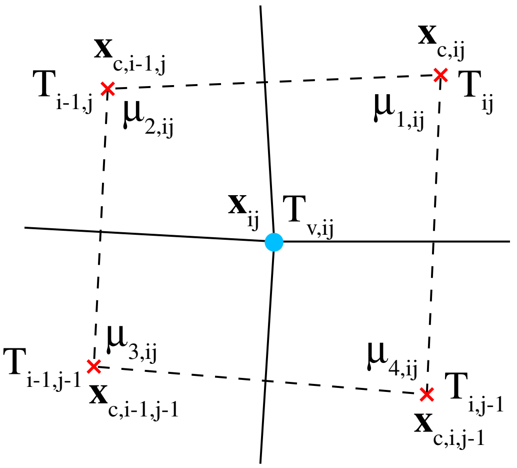

**Fig. 5.** Four-point stencil for evaluation of vertex temperatures $T_{v,ij}$. The $c$-quadrilateral is delineated with dashes.

which is symmetric with respect to cyclic permutations of the surrounding cell centers and second-order accurate. When combined with the finite-difference expression (32) for the fluxes $H_{ijm}$, it yields a nine-point stencil for the 2D spatial discretization of Eq. (2).

Assume for a moment that the conduction coefficient $\kappa$ is continuous across all the relevant cell faces and consider a $c$-quadrilateral made of vertices $\mathbf{x}_{c,ij}, \mathbf{x}_{c,i-1,j}, \mathbf{x}_{c,i,j-1}$ and $\mathbf{x}_{c,i-1,j-1}$ (see Fig. 5). An effective way to construct the required interpolation is to use a bilinear mapping of the $c$-quadrilateral onto a standard $(\xi,\eta) \in [-1,+1] \times [-1,+1]$ square in the computational space [10, §11.4], which is given by

$$\mathbf{x} = \frac{1}{4}\left[\mathbf{x}_{c,ij}(1+\xi)(1+\eta) + \mathbf{x}_{c,i-1,j}(1-\xi)(1+\eta) + \mathbf{x}_{c,i,j-1}(1+\xi)(1-\eta) + \mathbf{x}_{c,i-1,j-1}(1-\xi)(1-\eta)\right]. \tag{36}$$

The required temperature interpolation can then be written as

$$T(\mathbf{x}) = \frac{1}{4}\left[T_{ij}(1+\xi)(1+\eta) + T_{i-1,j}(1-\xi)(1+\eta) + T_{i,j-1}(1+\xi)(1-\eta) + T_{i-1,j-1}(1-\xi)(1-\eta)\right]. \tag{37}$$

Clearly, this interpolation possesses the desired symmetry and is linear with respect to the physical coordinates $\mathbf{x}$ along all the four edges of the $c$-quadrilateral. More generally, if the original temperature distribution is an arbitrary linear function $T(\mathbf{x}) = a + \mathbf{b}\cdot\mathbf{x}$, the interpolation scheme (Eqs. (36) and (37)) reconstructs this function exactly. The latter fact actually means that our algorithm should reproduce exactly steady-state linear solutions of the diffusion equation on all types of distorted grids, and that it should be of second-order when converging to non-linear steady-state solutions.

Eqs. (36) and (37) dictate the following algorithm for evaluating $T_{v,ij}$: (i) set $\mathbf{x} = \mathbf{x}_{ij}$ on the left-hand side of Eq. (36) and calculate the corresponding natural coordinates $(\xi_{v,ij}, \eta_{v,ij})$ of vertex $(i,j)$ (here one has to solve a quadratic equation), then (ii) substitute $(\xi_{v,ij}, \eta_{v,ij})$ for $(\xi,\eta)$ in Eq. (37) and calculate $T_{v,ij}$. Generalization to a discontinuous conduction coefficient $\kappa$ is accomplished by referring to a corresponding 1D treatment (see, for example, [10, §3.3]) along each of the four edges of the $c$-quadrilateral. First of all note that the mapping (Eqs. (36) and (37)) becomes an exact 1D linear interpolation between two neighbor cell-centered temperatures $T_{ij}$ whenever vertex $\mathbf{x}_{ij}$ lies on any of the four edges of the $c$-quadrilateral. It is an elementary exercise to show that in this 1D case the correct interface (i.e. vertex) temperature, consistent with the continuity of the flux $\kappa\nabla T$, is obtained after the weight coefficients for $T_{ij}$ in the corresponding linear interpolation are multiplied by $\kappa_{ij}$. Hence, an obvious generalization to the 2D case will be to use in Eq. (35) the interpolation coefficients $\mu_{\alpha,ij}$, $\alpha = 1,2,3,4$, defined as

$$\mu_{\alpha,ij} = \beta_{\alpha,ij}\left(\sum_{\alpha=1}^{4}\beta_{\alpha,ij}\right)^{-1}, \tag{38}$$

$$\begin{aligned}
\beta_{1,ij} &= \kappa_{ij}(1+\xi_{v,ij})(1+\eta_{v,ij}),\\
\beta_{2,ij} &= \kappa_{i-1,j}(1-\xi_{v,ij})(1+\eta_{v,ij}),\\
\beta_{3,ij} &= \kappa_{i-1,j-1}(1-\xi_{v,ij})(1-\eta_{v,ij}),\\
\beta_{4,ij} &= \kappa_{i,j-1}(1+\xi_{v,ij})(1-\eta_{v,ij}).
\end{aligned} \tag{39}$$

Since temperature is inherently non-negative, we would like all the coefficients $\mu_{\alpha,ij}$ in Eq. (35) to be non-negative. However, once our grid becomes *strongly* distorted – i.e. we calculate either $|\xi_{v,ij}| > 1$ or $|\eta_{v,ij}| > 1$ from Eq. (36) – at least some of $\mu_{\alpha,ij}$ become negative. Then we face a dilemma: (i) either impose an artificial constraint $\mu_{\alpha,ij} \geqslant 0$ and loose the second-order convergence, or (ii) risk large numerical errors and negative temperatures with non-linear temperature profiles. Numerical tests with non-steady non-linear problems (see Section 5.4 below) compel us to decide for option (i).

Finally, substituting Eq. (35) into Eq. (32) and differentiating with respect to the corresponding temperatures, we obtain explicit expressions for the SSI coefficients $a_{ijm}$ and $b_{ijm}$,

$$a_{ijm} = \frac{\kappa_{f,ijm} R_{f,ijm}}{|\mathbf{l}_{v,ijm} \times \mathbf{l}_{c,ijm}|} \left[ |\mathbf{l}_{v,ijm}|^2 + (\mathbf{l}_{v,ijm} \cdot \mathbf{l}_{c,ijm}) \cdot \left\{ \begin{matrix} (\mu_{1,ij} - \mu_{2,i+1,j}), & m = 1, \\ (\mu_{1,ij} - \mu_{4,i,j+1}), & m = 2, \end{matrix} \right\} \right], \tag{40}$$

$$b_{ijm} = \frac{\kappa_{f,ijm} R_{f,ijm}}{|\mathbf{l}_{v,ijm} \times \mathbf{l}_{c,ijm}|} \left[ |\mathbf{l}_{v,ijm}|^2 + (\mathbf{l}_{v,ijm} \cdot \mathbf{l}_{c,ijm}) \cdot \left\{ \begin{matrix} (\mu_{3,i+1,j} - \mu_{4,i,j}), & m = 1, \\ (\mu_{3,i,j+1} - \mu_{2,i,j}), & m = 2, \end{matrix} \right\} \right]. \tag{41}$$

It may be noted here that a possible alternative to the bilinear mapping (Eqs. (36) and (37)) would be to use a least square approximation to the linear interpolation $T(\mathbf{x}) = a + \mathbf{b} \cdot \mathbf{x}$. An obvious shortcoming of this approach is that we loose the exact linear interpolation along the edges of the $c$-quadrilateral. We did not explore possible advantages of this option simply because good accuracy had already been achieved with the present method.

### 4.4. Face-centered conduction coefficients

Since our principal goal is to obtain a fast and economical numerical scheme, we want to avoid iterative solution of a large non-linear system of equations for the new temperatures $\widetilde{T}_{ij}$ required in the fully non-linear approach. Consequently, we are forced to use the old (i.e. from the previous time step) values of the conduction coefficient $\kappa_{ij}$. Within this approach we consider two options for interpolation between the cell-centered $\kappa_{ij}$ to obtain face-centered values $\kappa_{f,ijm}$. The first is a weighted arithmetic mean,

$$\kappa_{fa,ijm} = \frac{A^{-}_{\Delta,ijm}}{A^{-}_{\Delta,ijm} + A^{+}_{\Delta,ijm}} \kappa_{ij} + \frac{A^{+}_{\Delta,ijm}}{A^{-}_{\Delta,ijm} + A^{+}_{\Delta,ijm}} \cdot \left\{ \begin{matrix} \kappa_{i,j-1}, & m = 1, \\ \kappa_{i-1,j}, & m = 2, \end{matrix} \right\}, \tag{42}$$

where the weights are proportional to the areas $A^{\pm}_{\Delta,ijm}$ of the two triangles adjacent to face $(i,j,m)$; see Fig. 3. It is a simple linear interpolation along the normal direction $\mathbf{n}_{f,ijm}$.

The second option is derived from the self-consistent treatment of discontinuous $\kappa$ and leads to the weighted harmonic mean

$$\begin{aligned} \kappa_{fh,ij1} &= \frac{\kappa_{ij}\kappa_{i,j-1}\left(A^{-}_{\Delta,ij1} + A^{+}_{\Delta,ij1}\right)}{\kappa_{ij}A^{-}_{\Delta,ij1} + \kappa_{i,j-1}A^{+}_{\Delta,ij1}} = \frac{\kappa_{ij}\kappa_{i,j-1}}{\kappa_{fa,ij1}}, \\ \kappa_{fh,ij2} &= \frac{\kappa_{ij}\kappa_{i-1,j}\left(A^{-}_{\Delta,ij2} + A^{+}_{\Delta,ij2}\right)}{\kappa_{ij}A^{-}_{\Delta,ij2} + \kappa_{i-1,j}A^{+}_{\Delta,ij2}} = \frac{\kappa_{ij}\kappa_{i-1,j}}{\kappa_{fa,ij2}}, \end{aligned} \tag{43}$$

which is usually considered to be the preferred version of $\kappa_{f,ijm}$ [10, Chapter 3]. However, it is quite clear that, because we use the old values of $\kappa_{ij}$, the harmonic mean cannot be taken literally for simulating non-linear heat waves, which propagate into a cold medium with initially zero (or close to zero) temperature and conduction coefficient. Here, again we face a dilemma of either (i) to use the arithmetic mean (Eq. (42)) and loose the possibility to reproduce exactly piecewise linear solutions with $\kappa$ jumps, or (ii) to impose an ad hoc limit on possible cell-to-face variations of the conduction coefficient, such as

$$\kappa_{f,ijm} = \max\{\kappa_{f0,ijm}, \kappa_{fh,ijm}\}, \tag{44}$$

where

$$\kappa_{f0,ijm} = \delta_{\kappa f0} \cdot \left\{ \begin{matrix} \max\{\kappa_{ij}, \kappa_{i,j-1}\}, & m = 1, \\ \max\{\kappa_{ij}, \kappa_{i-1,j}\}, & m = 2, \end{matrix} \right. \tag{45}$$

and $\delta_{\kappa f0} \ll 1$ is a user-defined small parameter. And again, tests with non-linear heat waves in Sections 5.4 and 5.5 compel us to opt for (i).

### 4.5. Boundary conditions

We consider only the Dirichlet (specified temperature) and the Neumann (specified flux) boundary conditions that are relevant for material (electron or molecular) thermal conduction. Probably the easiest way to set up a Dirichlet boundary condition is to add an extra layer of virtual ("ghost") grid cells and assign the boundary temperature values $T_{bc,ij}$ to their centers. Also, cell-centered boundary values of the conduction coefficient $\kappa_{bc,ij}$ should be specified in this case.

For a Neumann boundary condition, corresponding boundary fluxes can be assigned directly, $H_{ijm} = H_{bc,ijm}$, at cell faces $(i,j,m)$ along the boundary. Note that in this case we still need the cell-centered boundary temperatures $T_{bc,ij}$ in the ghost cells to calculate the vertex temperatures $T_{v,ij}$ along the boundary; the latter are needed to calculate the fluxes across the neighboring non-boundary faces $(i,j,3-m)$ on non-orthogonal grids; see Eq. (32). The ghost-cell temperatures $T_{bc,ij}$ are simply taken from the neighboring physical cells. In this way also the symmetry (reflective) boundaries are automatically accounted for.

In addition, for all types of boundaries, we have to set $b_{ijm} = 0, \chi_{ijm} = 1$ at boundary faces $(i,j,m)$ where the unit normal $\mathbf{n}_{f,ijm}$ points inside the computational domain, and $a_{ijm} = 0, \chi_{ijm} = 0$ at boundary faces $(i,j,m)$ where $\mathbf{n}_{f,ijm}$ points outside the computational domain. Along the Neumann-type boundaries, where the fluxes $H_{ijm}$ are fixed by the boundary condition, we set both $a_{ijm} = b_{ijm} = 0$.

### 4.6. Time step control

The SSI algorithm requires a separate control of the time step $\Delta t$ to ensure convergence and accuracy of solution [7]. Because the energy correction $\delta_{ij}$ in Eq. (16) is taken from the previous step and cannot be reduced in the current cycle, we need two separate constraints on $\Delta t$ with two independent control parameters $\varepsilon_0$ and $\varepsilon_1$. Similar to Ref. [7], these conditions can be written as

$$\left|\frac{(H_{ij1} + H_{ij2} - H_{i,j+1,1} - H_{i+1,j,2} + q_{ij}M_{ij})\Delta t}{c_{V,ij}M_{ij} + (a_{ij1} + a_{ij2} + b_{i,j+1,1} + b_{i+1,j,2})\Delta t}\right| \leqslant (\varepsilon_0 - \varepsilon_1)(T_{ij} + T_s), \tag{46}$$

$$\left|\frac{\tilde{\delta}_{ij}}{c_{V,ij}M_{ij}}\right| \leqslant \varepsilon_1(T_{ij} + T_s), \tag{47}$$

where $T_s > 0$ is a problem-specific "sensitivity" threshold for temperature variations. Clearly, we must always choose $\varepsilon_1 < \varepsilon_0$.

Condition (47) guarantees that in the next cycle the relative temperature variation due to the SSI energy correction $\delta_{ij}$ will not exceed $\varepsilon_1$ for any new value of $\Delta t > 0$. When applied together at each time step, the two conditions (46) and (47) guarantee that the total relative temperature variation $|\tau_{ij}|/(T_{ij} + T_s)$ [where $\tau_{ij}$ is given by Eq. (16)] never exceeds $\varepsilon_0$. Note that, by reducing the current time step $\Delta t$, both conditions (46) and (47) can always be satisfied for any $\varepsilon_1 > 0$ and $\varepsilon_0 > \varepsilon_1$.

## 5. Numerical tests

We explore the properties of our algorithm by running two groups of tests. The first group is a selection of standard tests against simple steady-state analytical solutions, used in many previous publications [1,4–6]. These tests reveal the basic properties of our algorithm with respect to spatial discretization. The second group includes three time-dependent problems, which demonstrate the basic features of the SSI method.

All the test problems are simulated on a standard set of four different grids in a unit square $(x,y) \in [0,1] \times [0,1]$ that are successively refined by decreasing the parameter

$$h = \left(n_x n_y\right)^{-1/2}, \tag{48}$$

where $n_x$ and $n_y$ are, respectively, the numbers of cells along the $x$ and $y$-axes. Usually we have $n_x = n_y$, so that $h = 1/n_x$. The four standard grids that we use below are shown in Fig. 6. The smooth "wavy" grid in Fig. 6(c) is constructed by applying a diagonal shift [5]

$$\begin{aligned} x'_{ij} &= x_{ij} + a_0 \sin(x_{ij}) \cos(y_{ij}), \\ y'_{ij} &= y_{ij} + a_0 \sin(x_{ij}) \cos(y_{ij}), \end{aligned} \tag{49}$$

to a square grid of Fig. 6(a) with $a_0 = 0.1$. The random grid in Fig. 6(d) is obtained by shifting each inner vertex of the square grid to a random position on a circle of radius $0.2h$ around the original vertex location [4]. Note that of these four grids only the Kershaw grid is strongly distorted in the sense that some of the interpolation weights $(1 \pm \xi_{v,ij})(1 \pm \eta_{v,ij})$ in Eq. (37) become negative.

As usual, we introduce two norms for truncation errors

$$\delta T^h_m = \max_{(i,j)} \left|T_{ij} - T^c_{ij}\right|, \quad \delta T^h_{L2} = \left[\sum_{i,j} \left(T_{ij} - T^c_{ij}\right)^2 V_{ij}\right]^{1/2}, \tag{50}$$

to investigate the convergence of the numerical scheme; here $T^c_{ij}$ is the value of the exact solution $T(x,y)$ at the midpoint $(x,y) = (x_{c,ij}, y_{c,ij})$ of cell $(i,j)$, and $V_{ij}$ is the volume (area) of this cell. Correspondingly, we have two values

$$q_m = \frac{\log\left(\delta T^{h_1}_m / \delta T^{h_2}_m\right)}{\log(h_1/h_2)}, \quad q_{L2} = \frac{\log\left(\delta T^{h_1}_{L2} / \delta T^{h_2}_{L2}\right)}{\log(h_1/h_2)}, \tag{51}$$

for the spatial convergence order of the scheme.

To reach numerical steady states in steady-state tests, we applied appropriate boundary conditions, a certain initial condition (which typically was $T(x,y) = 0$), and ran the code until the errors ceased to change in time. It was explicitly checked that the final results did not depend on the initial conditions, and only the number of required time steps changed.

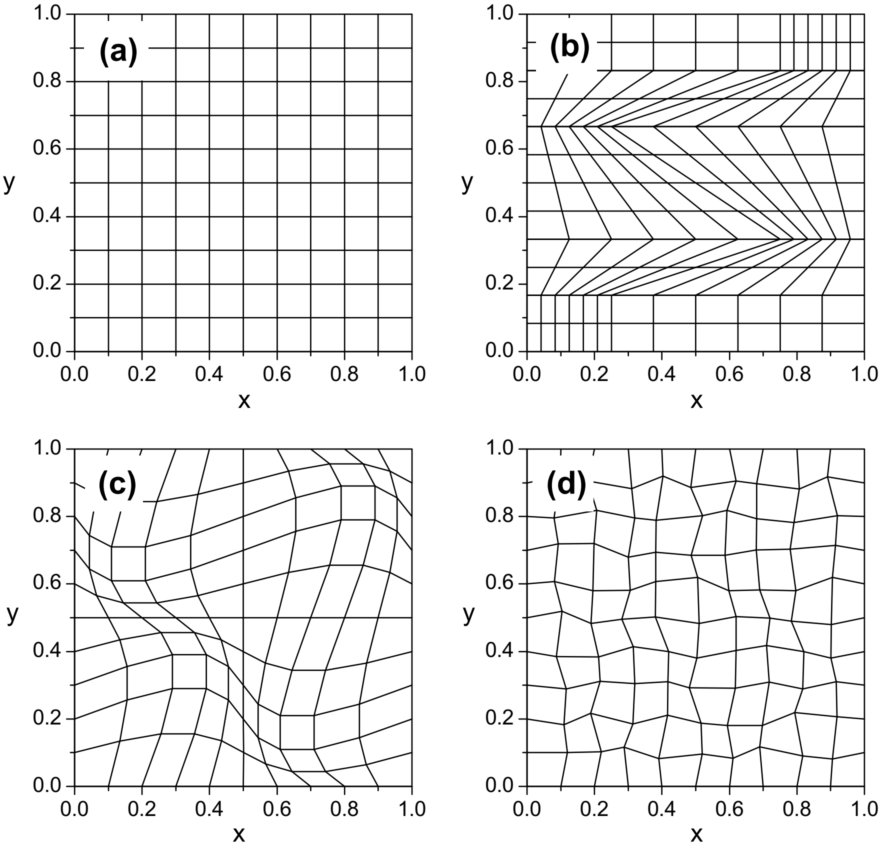

**Fig. 6.** Four standard grids on a unit square: (a) orthogonal square grid with $n_x = n_y = 10$; (b) Kershaw grid with $n_x = n_y = 12$; (c) smooth "wavy" grid, obtained by transformation (49), with $n_x = n_y = 10$ and (d) random grid with $n_x = n_y = 10$.

### 5.1. Steady-state linear solution

First of all, we test our algorithm against the linear steady-state solution

$$T(x,y) = x \tag{52}$$

in the unit square $0 \leqslant x,\ y \leqslant 1$, obtained with the boundary conditions $T(0,y) = 0,\ T(1,y) = 1$ and $\partial T/\partial y = 0$ along $y = 0$ and $y = 1$; here the values $\rho c_V = 1,\ \kappa = 1,\ Q = 0$ are assumed. Once the steady-state is reached, our numerical solution reproduces the analytical one exactly on all four grids shown in Fig. 6. In this respect, our scheme is on a par with those proposed in Refs. [4,5], and superior to the one proposed in Ref. [6]. Note that, in order to recover the exact linear solution on the strongly distorted Kershaw mesh, we have to allow negative values of the interpolation weights $(1 \pm \xi)(1 \pm \eta)$ in Eq. (37). When, however, these weights are constrained to be non-negative, we get finite truncation errors with the first-order convergence rate.

Because we use a semi-implicit rather than fully implicit method for time discretization, the linear solution provides a good opportunity to study the temporal convergence of our scheme. In general, the efficiency of temporal convergence depends on the type of grid and the initial condition. Fig. 7 illustrates the convergence for a difficult case of the random grid with 100 × 100 cells and the initial condition $T(x,y) = 0$ at $t = 0$. If we use a fixed value of the time step $\Delta t$ (solid curves in Fig. 7), then a rapid convergence (on a physical relaxation timescale $t_{rel} \approx 0.1$) to the steady-state solution is achieved with $\Delta t \leqslant 0.001$, but no convergence is observed for $\Delta t \geqslant 0.002$. Although absence of convergence for large $\Delta t$ looks more like zero stability than instability (the errors saturate at a finite level), this example indicates that the original claim by Livne and Glasner [7] of the SSI method being unconditionally stable might in fact be too strong and somewhat misleading for practical applications. The conclusion is that, in the SSI method, a special care should be taken of the time step control. Note that the convergence threshold $\Delta t \approx 2 \times 10^{-3}$ in the above example is still significantly larger than the stability threshold $\Delta t \leqslant h^2/(2\kappa) = 5 \times 10^{-5}$ of the explicit scheme.

At the same time, as it is illustrated with the dotted curve in Fig. 7, the "dynamic" time-step constraints (46) and (47), based on accuracy considerations, prove to be quite sufficient for practical needs even with moderate values of $\varepsilon_0 = 0.2, \varepsilon_1 \lesssim \varepsilon_0/4$. Although these conditions do not ensure full convergence to the steady-state solution, they suppress

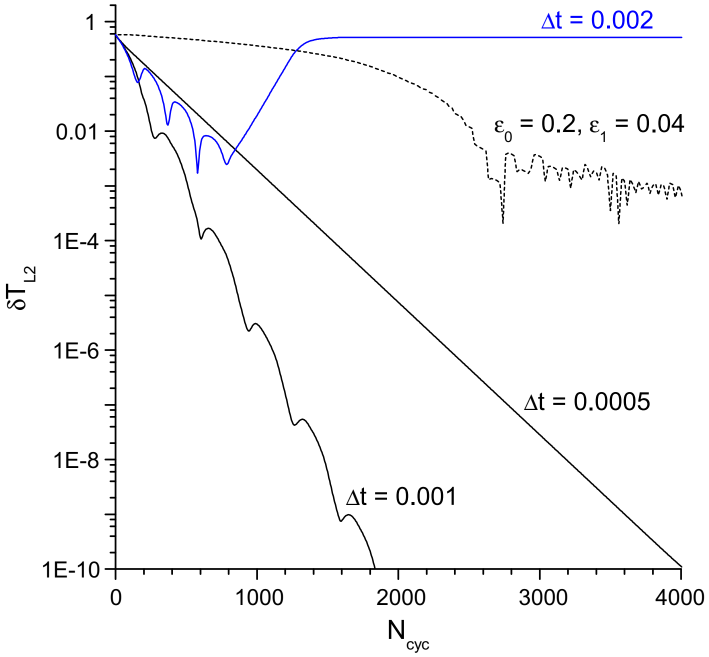

**Fig. 7.** The $L_2$ numerical error $\delta T_{L2}$ as a function of the number of time steps $N_{cyc}$ for the linear problem (52) on the random grid with $n_x = n_y = 100$. The three solid curves have been computed with three different fixed values of the time step $\Delta t$. The dotted curve has been obtained by applying the "dynamic" criterion (46) with the values of $\varepsilon_0 = 0.2,\ \varepsilon_1 = 0.04,\ T_s = 10^{-3}$.

the numerical errors to a low enough level of $\delta T_m, \delta T_{L2} \lesssim 1\text{–}2 \times 10^{-3}$ that should be satisfactory for most practical applications. The criterion based on Eqs. (46) and (47) does not lead to full convergence because, as the temperature variations become smaller and smaller, it allows larger and larger time steps, which finally surpass the convergence threshold.

### *5.2. Steady-state non-linear solution*

Following Refs. [4–6], we take a 1D steady-state non-linear solution

$$T(x,y) = a + bx + cx^4, \tag{53}$$

where $a$, $b$ and $c$ are constants, to verify the convergence of our scheme with respect to spatial discretization. This solution is obtained with the source term

$$Q = Q(x,y) = x^2, \tag{54}$$

and the boundary conditions of a zero normal flux, $\partial T/\partial y = 0$, at $y = 0$ and 1, and of fixed temperatures,

$$T(0,y) = a, \quad T(1,y) = a + b + c, \tag{55}$$

at $x = 0$ and $x = 1$.

Table 1 compares the convergence results for our scheme with those from Ref. [5], obtained with the values of

$$a = \frac{1+8\kappa}{6(1+4\kappa)}, \quad b = \frac{1+8\kappa}{12\kappa(1+4\kappa)}, \quad c = -\frac{1}{12\kappa}, \quad \kappa = \frac{1}{30} \tag{56}$$

**Table 1**
Steady-state non-linear solution on a random grid: comparison with the algorithm of Shashkov and Steinberg [5].

| Algorithm | $n_x = n_y$ | $\delta T_m$ | $\delta T_{L2}$ | $q_m$ | $q_{L2}$ |
|---|---|---|---|---|---|
| Ref. [5] | 10 | $4.34 \times 10^{-2}$ | $1.87 \times 10^{-2}$ | 2.04 | 2.20 |
| | 20 | $1.05 \times 10^{-2}$ | $4.06 \times 10^{-3}$ | 1.72 | 1.79 |
| | 40 | $3.18 \times 10^{-3}$ | $1.17 \times 10^{-3}$ | – | – |
| This work | 10 | $4.01 \times 10^{-2}$ | $1.86 \times 10^{-2}$ | 1.67 | 1.96 |
| | 20 | $1.26 \times 10^{-2}$ | $4.79 \times 10^{-3}$ | 1.92 | 1.97 |
| | 40 | $3.32 \times 10^{-3}$ | $1.22 \times 10^{-3}$ | 1.97 | 2.02 |
| | 80 | $8.48 \times 10^{-4}$ | $3.01 \times 10^{-4}$ | – | – |

**Table 2**
Steady-state non-linear solution on a random grid: comparison with the algorithm of Breil and Maire [6].

| Grid/algorithm | $n_x = n_y$ | $\delta T_m$ | $\delta T_{L2}$ | $q_m$ | $q_{L2}$ |
|---|---|---|---|---|---|
| Kershaw/Ref. [6] | 6 | $4.29 \times 10^{-2}$ | $1.86 \times 10^{-2}$ | 1.5 | 1.84 |
| | 18 | $8.17 \times 10^{-3}$ | $2.44 \times 10^{-3}$ | 1.66 | 1.93 |
| | 36 | $2.60 \times 10^{-3}$ | $6.43 \times 10^{-4}$ | – | – |
| Kershaw/this work | 6 | $4.84 \times 10^{-3}$ | $2.23 \times 10^{-3}$ | 1.0 | 1.18 |
| | 18 | $1.62 \times 10^{-3}$ | $6.07 \times 10^{-4}$ | 1.23 | 1.42 |
| | 36 | $6.90 \times 10^{-4}$ | $2.27 \times 10^{-4}$ | 1.58 | 1.75 |
| | 72 | $2.30 \times 10^{-4}$ | $6.75 \times 10^{-5}$ | – | – |
| Wavy/Ref. [6] | 10 | $6.17 \times 10^{-3}$ | $2.49 \times 10^{-3}$ | 1.31 | 1.84 |
| | 20 | $1.90 \times 10^{-3}$ | $6.93 \times 10^{-4}$ | 1.87 | 1.96 |
| | 40 | $5.19 \times 10^{-4}$ | $1.78 \times 10^{-4}$ | – | – |
| Wavy/this work | 10 | $2.33 \times 10^{-3}$ | $8.06 \times 10^{-4}$ | 1.75 | 1.95 |
| | 20 | $6.92 \times 10^{-4}$ | $2.09 \times 10^{-4}$ | 1.86 | 1.99 |
| | 40 | $1.90 \times 10^{-4}$ | $5.26 \times 10^{-5}$ | 1.93 | 1.99 |
| | 80 | $4.97 \times 10^{-5}$ | $1.32 \times 10^{-5}$ | – | – |
| Random/Ref. [6] | 10 | $1.92 \times 10^{-3}$ | $6.21 \times 10^{-4}$ | 0.37 | 0.69 |
| | 20 | $1.48 \times 10^{-3}$ | $3.84 \times 10^{-4}$ | 0.18 | 0.15 |
| | 40 | $1.31 \times 10^{-3}$ | $3.46 \times 10^{-4}$ | – | – |
| Random/this work | 10 | $1.34 \times 10^{-3}$ | $6.20 \times 10^{-4}$ | 1.67 | 1.95 |
| | 20 | $4.20 \times 10^{-4}$ | $1.60 \times 10^{-4}$ | 1.92 | 1.98 |
| | 40 | $1.11 \times 10^{-4}$ | $4.06 \times 10^{-5}$ | 1.97 | 2.02 |
| | 80 | $2.83 \times 10^{-5}$ | $1.00 \times 10^{-5}$ | – | – |

on a random grid of Fig. 6(d). Similar to the support-operators method of Ref. [5], our scheme demonstrates a second-order convergence rate in both norms, and nearly the same absolute values of the truncation errors.

A more detailed comparison with the recently published work of Breil and Maire [6] on three types of distorted grids is presented in Table 2. In this case the values of

$$a = 0, \quad b = 1 + \frac{1}{12\kappa}, \quad c = -\frac{1}{12\kappa}, \quad \kappa = 1 \tag{57}$$

are chosen. Fig. 8 shows how the $L_2$ error $\delta T_{L2}$ decreases with the decreasing mesh size $h$ on the four grids of Fig. 6. No constraints to ensure positiveness of $\mu_{\alpha,ij}$ in Eq. (35) have been imposed here. Clearly, for all the four grids the second-order asymptotical convergence is observed. In this respect our scheme proves to be more efficient than that of Ref. [6], where slow (if any) convergence was observed on the random mesh. In addition, as it is seen from Table 2, in all cases considered our errors $\delta T_m, \delta T_{L2}$ are significantly lower than those in Ref. [6]. Note that in our case the random grid errors are practically equal to those on the square grid (see Fig. 8).

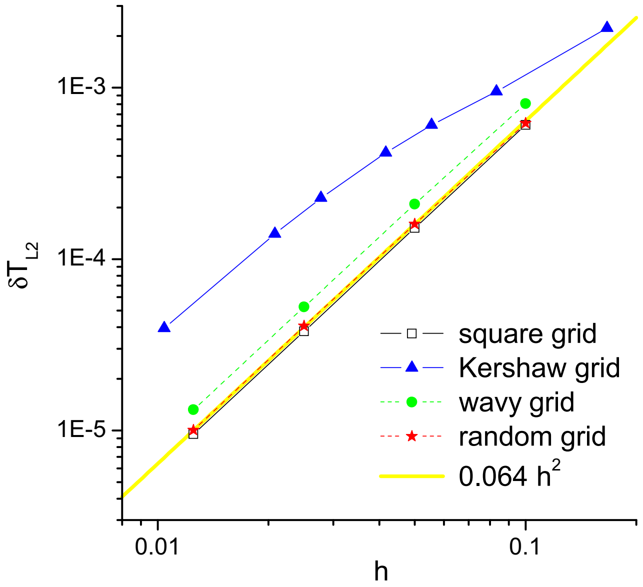

**Fig. 8.** The $L_2$ numerical error $\delta T_{L2}$ as a function of mesh size $h$ for the non-linear problems (53)–(55), (57) on the four grids of Fig. 6.

### 5.3. Steady-state solutions with discontinuous conduction coefficient

Here, we perform convergence analysis for a steady-state solution of the form [5]

$$T(x,y)=\begin{cases} a+bx+cy, & 0\leqslant x\leqslant 0.5,\\ a+b\frac{\kappa_2-\kappa_1}{2\kappa_2}+b\frac{\kappa_1}{\kappa_2}x+cy, & 0.5\leqslant x\leqslant 1, \end{cases} \tag{58}$$

where the conduction coefficient

$$\kappa=\begin{cases} \kappa_1, & 0\leqslant x<0.5,\\ \kappa_2, & 0.5<x\leqslant 1, \end{cases} \tag{59}$$

has a jump at $x=0.5$. For $c\neq 0$ we obtain a solution with a discontinuous tangential flux, proposed in Ref. [5] as a special test case. To numerically simulate the above solution, we applied the Dirichlet boundary condition (specified $T$) as dictated by Eq. (58) along the entire perimeter of the computational domain $0\leqslant x,y\leqslant 1$.

As might be expected, our algorithm reproduces the present solution exactly only when a harmonic-mean formula (43) is used for the face-centered conduction coefficient $\kappa_f$, and only on rectangular grids, where the natural $\xi,\eta$ coordinates in the bilinear interpolation (37) are proportional to the physical $x,y$ coordinates. On distorted grids it is of the first-order convergence rate. In this respect our scheme is clearly inferior to those from Refs. [4,5].

Table 3 presents the results, obtained with $a=c=0, b=2\kappa_2/(\kappa_1+\kappa_2), \kappa_1=1, \kappa_2=9$ for a solution with continuous tangential flux, whereas Table 4 shows a comparison with Ref. [6] for a solution with discontinuous tangential flux, obtained with $a=b=c=1, \kappa_1=1, \kappa_2=4$. Two types of non-orthogonal grids have been tested, namely, the wavy grid (Fig. 6(c)) and the random grid (Fig. 6(d)). As proposed in Ref. [5], the coordinate line $x=0.5$ has been made straight on the random grid.

Tables 3 and 4 demonstrate that, despite only a first-order convergence rate, our scheme generates rather low absolute errors on moderately distorted grids, which makes it quite satisfactory for practical applications. The results are particularly good for the random grid – in contrast to the algorithm of Ref. [6], for which the random grid appears to be the most difficult case. However, our scheme becomes inferior to that from Ref. [6] on a fine wavy grid, where the algorithm of Ref. [6] demonstrates an almost second-order convergence rate.

**Table 3**
Steady-state piecewise-linear solution with discontinuous conduction coefficient.

| | $n_x=n_y$ | $\delta T_m$ | $\delta T_{L2}$ | $q_m$ | $q_{L2}$ |
|---|---|---|---|---|---|
| Wavy grid | 10 | $5.35\times10^{-3}$ | $1.55\times10^{-3}$ | 0.48 | 0.70 |
| | 20 | $3.83\times10^{-3}$ | $9.55\times10^{-4}$ | 0.71 | 0.87 |
| | 40 | $2.34\times10^{-3}$ | $5.22\times10^{-4}$ | 0.86 | 0.94 |
| | 80 | $1.29\times10^{-3}$ | $2.72\times10^{-4}$ | – | – |
| Random grid | 10 | $1.27\times10^{-3}$ | $2.19\times10^{-4}$ | 0.67 | 1.17 |
| | 20 | $7.96\times10^{-4}$ | $9.75\times10^{-5}$ | 1.49 | 1.81 |
| | 40 | $2.84\times10^{-4}$ | $2.79\times10^{-5}$ | 0.36 | 0.94 |
| | 80 | $2.21\times10^{-4}$ | $1.45\times10^{-5}$ | – | – |

**Table 4**
Steady-state piecewise-linear solution with discontinuous tangential flux: comparison with the algorithm of Breil and Maire [6].

| Grid/algorithm | $n_x=n_y$ | $\delta T_m$ | $\delta T_{L2}$ | $q_m$ | $q_{L2}$ |
|---|---|---|---|---|---|
| Wavy/Ref. [6] | 10 | $1.32\times10^{-2}$ | $4.06\times10^{-3}$ | 1.83 | 1.83 |
| | 20 | $3.72\times10^{-3}$ | $1.14\times10^{-3}$ | 1.94 | 1.95 |
| | 40 | $9.69\times10^{-4}$ | $2.96\times10^{-4}$ | – | – |
| Wavy/this work | 10 | $7.73\times10^{-3}$ | $1.41\times10^{-3}$ | 0.64 | 0.79 |
| | 20 | $4.97\times10^{-3}$ | $8.16\times10^{-4}$ | 0.85 | 0.90 |
| | 40 | $2.76\times10^{-3}$ | $4.36\times10^{-4}$ | 0.95 | 0.94 |
| | 80 | $1.43\times10^{-3}$ | $2.27\times10^{-4}$ | – | – |
| Random/Ref. [6] | 10 | $6.71\times10^{-3}$ | $2.03\times10^{-3}$ | 0.67 | 0.87 |
| | 20 | $4.21\times10^{-3}$ | $1.11\times10^{-3}$ | 0.5 | 0.77 |
| | 40 | $2.97\times10^{-3}$ | $6.49\times10^{-4}$ | – | – |
| Random/this work | 10 | $3.85\times10^{-4}$ | $7.83\times10^{-5}$ | 1.10 | 1.91 |
| | 20 | $1.79\times10^{-4}$ | $2.08\times10^{-5}$ | 0.95 | 1.08 |
| | 40 | $9.28\times10^{-5}$ | $9.82\times10^{-6}$ | 0.87 | 1.62 |
| | 80 | $5.06\times10^{-5}$ | $3.20\times10^{-6}$ | – | – |

### 5.4. Planar non-linear heat wave in warm medium

For practical applications, it is important to know how the numerical algorithm converges in time and space when strong heat waves are simulated. Here, we use a solution proposed in the original publication [7] on the SSI method, which describes a 1D non-linear heat wave

$$T(t,x) = 1 + \phi(\xi), \quad \xi = \kappa_0^{-1}(t - x), \tag{60}$$

propagating with a fixed (unit) velocity in a uniform medium with the initial temperature $T(-\infty, x) = 1$. For $\rho c_V = 1$ and a conduction coefficient of the form

$$\kappa = \kappa_0 T^4, \tag{61}$$

the function $\phi(\xi)$ is defined implicitly by the equation

$$\xi = \ln\phi + 4\phi + 3\phi^2 + \frac{4}{3}\phi^3 + \frac{1}{4}\phi^4. \tag{62}$$

As in the previous tests, we simulate this solution in the unit square $0 \leqslant x,\ y \leqslant 1$ by applying the boundary conditions of a zero normal flux, $\partial T/\partial y = 0$, along the $y = 0$ and $y = 1$ edges, and the exact values of $T(t,0)$ and $T(t,1)$ from Eq. (60) along the $x = 0$ and $x = 1$ edges. For time step control we use the "dynamic" criterion (46) and (47), which contains three control parameters $\varepsilon_0, \varepsilon_1$ and $T_s$. Once significantly below 1, the value of $T_s$ does not affect the results; in all the runs it was fixed at $T_s = 10^{-3}$. Of the remaining two, it is always $\varepsilon_1$ which limits the time step in the present problem – provided that a sufficiently large value of $\varepsilon_0 \gtrsim 2\varepsilon_1$ is chosen. The cause is a relatively high initial temperature $T = 1$, for which the relative temperature increments, that are to be constrained by $\varepsilon_0$, become automatically low once the constraint by $\varepsilon_1$ has been imposed. Therefore, we fixed the value of $\varepsilon_0 = 0.2$ and explored the range of $\varepsilon_1 \leqslant 0.1$.

In general, to obtain convergence to a non-steady exact solution, one has to reduce both the time step and the mesh size. In our case, time convergence cannot be faster than of the first-order. If we fix the mesh size and diminish the time step, the truncation errors saturate at a certain level. Evidently, this asymptotic level is determined by the spatial properties of the numerical algorithm and the grid. Of the three distorted grids in Fig. 6, our spatial algorithm is most accurate for the random grid, which is the least distorted.

The results of test runs on random grids are presented in Table 5 and Fig. 9 for the time $t = 0.8$. First of all, these data confirm the conclusion of Ref. [7] that for a good accuracy one has to use the values of $\varepsilon_1 \lesssim 0.02$, whereas with $\varepsilon_1 \geqslant 0.1$ the accuracy of the SSI method becomes rather poor. As it is seen from Table 5, error saturation occurs rather late, at time step values below those typically used. Hence, to improve the accuracy of simulation on a weakly distorted grid, one should in the first place try shorter time steps.

An important practical observation emerges after we compare the results obtained with different versions of the face-centered conduction coefficient $\kappa_f$ [cf. Eqs. (42) and (43)]. On random grids, the accuracy of the numerical solution improves considerably when the weighted arithmetic mean $\kappa_f$ is used instead of the weighted harmonic mean (recall that the latter is the text-book prescription for discontinuous conduction coefficients). As is clearly illustrated by the insert in Fig. 9, the front of the non-linear heat wave, calculated with the harmonic-mean $\kappa_f$, lags significantly behind the exact solution. Such a behavior manifests itself even more dramatically in the test problem of the next paragraph.

The relative degree of mesh distortion increases as we change from the random grid to the wavy grid, and from the wavy grid to the Kershaw grid (the smoothness of the grid appears to be relatively unimportant for our scheme). On the Kershaw grid we already have vertices which "stick out" of the $c$-quadrilateral in Fig. 5; for such vertices we get negative interpolation weights $\mu_{\alpha,ij}$ in Eq. (35). As it turns out, without constraining these weights to be non-negative, we cannot simulate the present test problem within a practically reasonable number of time steps of a few thousand. Thus, on strongly distorted grids we are compelled to sacrifice the second-order of spatial convergence for the sake of practicality and introduce an additional restriction $\mu_{\alpha,ij} \geqslant 0$ for the interpolation coefficients in Eq. (35). The ensuing loss of accuracy turns out to be practically insignificant. The results obtained in this manner are presented in Table 6 and Fig. 10.

One sees that numerical errors on the Kershaw grid are typically significantly higher than on the random grid. To reduce numerical errors, one should in the first place increase the spatial resolution. And no advantage is gained by using the

**Table 5**
Numerical errors $\delta T_m, \delta T_{L2}$ and the total number of time steps $N_{cyc}$ at $t = 0.8$ for the constant-speed heat wave problem on random grids. Values in parentheses have been obtained with the harmonic-mean option for $\kappa_f$.

| Grid | $\varepsilon_1$ | $\delta T_m$ | $\delta T_{L2}$ | $N_{cyc}$ |
|---|---|---|---|---|
| Random 40 × 40 | 0.10 | $1.92 \times 10^{-1}$ $(3.07 \times 10^{-1})$ | $8.26 \times 10^{-2}$ $(1.03 \times 10^{-1})$ | 284 (278) |
| | 0.02 | $4.63 \times 10^{-2}$ $(1.64 \times 10^{-1})$ | $1.70 \times 10^{-2}$ $(3.42 \times 10^{-2})$ | 653 (625) |
| | 0.005 | $2.15 \times 10^{-2}$ $(1.27 \times 10^{-1})$ | $5.55 \times 10^{-3}$ $(2.12 \times 10^{-2})$ | 1304 (1263) |
| Random 80 × 80 | 0.10 | $2.04 \times 10^{-1}$ | $7.07 \times 10^{-2}$ | 588 |
| | 0.02 | $4.26 \times 10^{-2}$ | $1.69 \times 10^{-2}$ | 1292 |
| | 0.005 | $1.01 \times 10^{-2}$ | $4.20 \times 10^{-3}$ | 2661 |

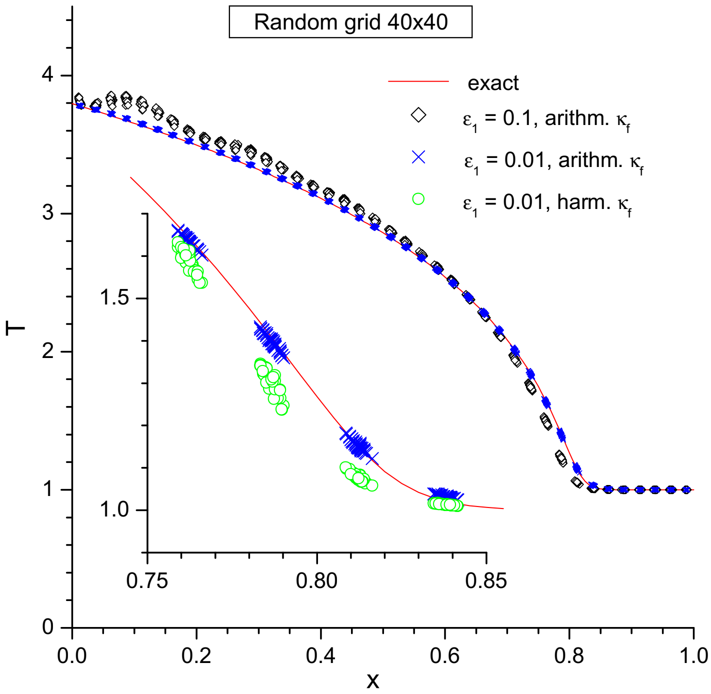

**Fig. 9.** Constant-speed heat wave in warm medium on a random grid with 40 × 40 cells at time $t = 0.8$. Diamonds and crosses: arithmetic-mean option for $\kappa_f$ with, respectively, $\varepsilon_1 = 0.1$ and 0.01; empty circles: harmonic-mean option for $\kappa_f$ with $\varepsilon_1 = 0.01$.

**Table 6**
Same as Table 5 but for the Kershaw grid.

| Grid | $\varepsilon_1$ | $\delta T_m$ | $\delta T_{L2}$ | $N_{cyc}$ |
|---|---|---|---|---|
| Kershaw 36 × 36 | 0.10 | $3.71 \times 10^{-1}$ $(3.51 \times 10^{-1})$ | $6.50 \times 10^{-2}$ $(6.03 \times 10^{-2})$ | 462 (453) |
| | 0.02 | $4.99 \times 10^{-1}$ $(4.75 \times 10^{-1})$ | $9.11 \times 10^{-2}$ $(7.51 \times 10^{-2})$ | 1064 (1040) |
| | 0.005 | $5.50 \times 10^{-1}$ $(5.29 \times 10^{-1})$ | $1.12 \times 10^{-1}$ $(9.40 \times 10^{-2})$ | 2215 (2177) |
| Kershaw 72 × 72 | 0.10 | $1.94 \times 10^{-1}$ | $4.96 \times 10^{-2}$ | 1041 |
| | 0.02 | $2.52 \times 10^{-1}$ | $3.03 \times 10^{-2}$ | 2240 |
| | 0.005 | $3.13 \times 10^{-1}$ | $4.28 \times 10^{-2}$ | 4638 |

arithmetic mean $\kappa_f$ instead of the harmonic mean: as it is seen in Fig. 10, both formulae give a relatively large and practically the same scatter of numerical points in the vicinity of the wave front. The results for the wavy grid lie between those for the random and the Kershaw grids. In all the test runs the SSI method allowed us to achieve acceptable error levels in about 1000–3000 time steps over the characteristic physical time scale.

### 5.5. Planar non-linear heat wave in a cold wall

As a show case for non-linear thermal waves, we consider a self-similar plane-parallel solution of Eq. (2) with

$$\kappa = \kappa_0 T^n, \tag{63}$$

which describes a heat wave launched into an initially cold $[T(0,x) = 0]$ half-space $x \geqslant 0$ by a fixed boundary temperature $T(t,0) = T_0$. Once we introduce dimensionless variables

$$\tau = \tau(\xi) = \frac{T}{T_0}, \quad \xi = \left(\frac{n+1}{2}\frac{\rho c_V}{\kappa_0 T_0^n}\right)^{1/2}\frac{x}{\sqrt{t}}, \tag{64}$$

Eq. (2) is reduced to an ordinary differential equation

$$\frac{d^2\tau^{n+1}}{d\xi^2} + \xi\frac{d\tau}{d\xi} = 0, \tag{65}$$

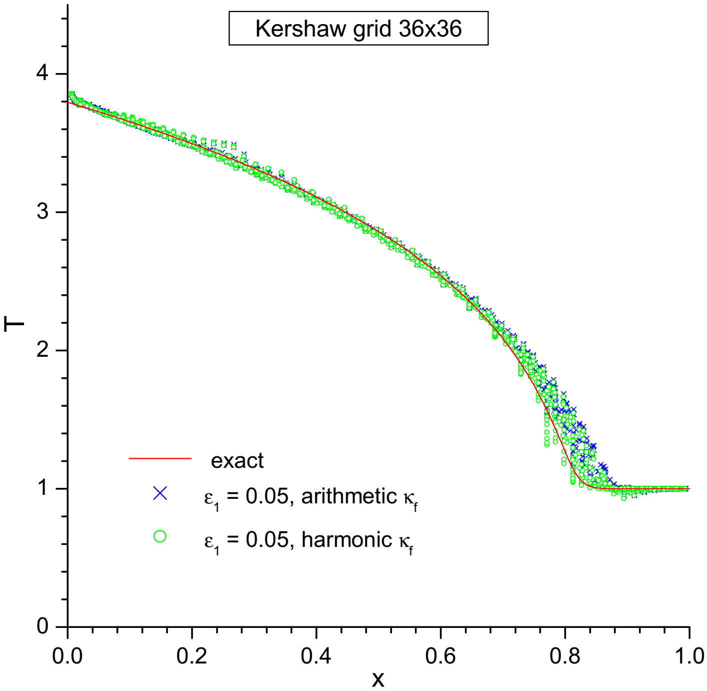

**Fig. 10.** Same as Fig. 9 but for a Kershaw grid with 36 × 36 cells and $\varepsilon_1 = 0.05$. Crosses: arithmetic mean $\kappa_f$; empty circles: harmonic mean $\kappa_f$.

which is to be solved with the boundary conditions $\tau = 1$ at $\xi = 0$, and $\tau = \tau^n(d\tau/d\xi) = 0$ at an unknown front position $\xi = \xi_0$; the latter must be determined in the process of solution of the boundary value problem. One can prove that at $\xi = \xi_0$ our solution has a singularity of the form $\tau \propto (\xi_0 - \xi)^{1/n}$. Here, we choose the value $n = 3$, for which $\xi_0 = 1.23117297028$.

To demonstrate the accuracy of our scheme for this problem, we performed 2D runs in the unit square $0 \leqslant x, y \leqslant 1$ on an orthogonal grid with $100 \times 100$ cells, and with the parameter values of $\kappa_0 = T_0 = \rho c_V = 1$. The boundary conditions $T(t, 0, y) = 1, T(t, 1, y) = 0$, and $\partial T/\partial y = 0$ at $y = 0$ and $y = 1$ were applied. Simulations were stopped at $t = 1$, when the wave front should arrive at $x = x_{f,ex} = 2^{-1/2}\xi_0 = 0.870571$. In this problem the time step $\Delta t$ is primarily limited by the values of $\varepsilon_0$ and $T_s$ in condition (46); the value of $\varepsilon_1$ is insignificant, provided that $0.1\varepsilon_0 \lesssim \varepsilon_1 \lesssim 0.5\varepsilon_0$. Excellent accuracy is achieved with the values of $\varepsilon_0 = 0.2$ and $T_s = 10^{-3}$; see Table 7.

The present problem has a difficulty of choosing an adequate definition for numerical error because of a singular behavior of the exact solution. Here, we simply compare the computed position of the wave front $x_f$ at $t = 1$ with the exact value $x_{f,ex} = 2^{-1/2}\xi_0$. The numerical value $x_f$ is reconstructed by performing a linear fit to the values of $T^3$ in the immediate vicinity of the front. Typical error of such reconstruction is about $\delta x_f \approx \pm 0.0005$.

The results presented in Fig. 11 and Table 7 demonstrate a dramatic difference between the arithmetic-mean and the harmonic-mean options for the face-centered conduction coefficient $\kappa_f$. While excellent results are obtained with the arithmetic mean, the harmonic mean actually fails to reproduce the vicinity of the wave front with the values $\delta_{\kappa f0} \ll 1$ of the jump limiting factor $\delta_{\kappa f0}$ in Eq. (45). To approach the exact solution, we have to go to the values $\delta_{\kappa f0} > 0.5$, for which the use of the

**Table 7**
Simulation results for the non-linear planar thermal wave on a square 100 × 100 grid. The computed wave front position $x_f$ is to be compared with the exact value $x_{f,ex} = 0.870571$.

| $\varepsilon_0$ | $\varepsilon_1$ | $T_s$ | $\delta_{\kappa f0}$ | $x_f$ | $N_{cyc}$ |
|---|---|---|---|---|---|
| *Arithmetic mean* $\kappa_f$ | | | | | |
| 0.5 | 0.05 | $10^{-3}$ | – | 0.8676 | 1704 |
| 0.2 | 0.02 | $10^{-3}$ | – | 0.8702 | 3053 |
| 0.05 | 0.005 | $10^{-3}$ | – | 0.8715 | 11,821 |
| 0.2 | 0.02 | $10^{-2}$ | – | 0.8683 | 1892 |
| *Harmonic mean* $\kappa_f$ | | | | | |
| 0.2 | 0.02 | $10^{-3}$ | 0.01 | 0.76 ($\pm$0.01) | 4510 |
| 0.2 | 0.02 | $10^{-3}$ | 0.1 | 0.845 ($\pm$0.005) | 3424 |
| 0.2 | 0.02 | $10^{-3}$ | 0.5 | 0.866 ($\pm$0.001) | 3010 |

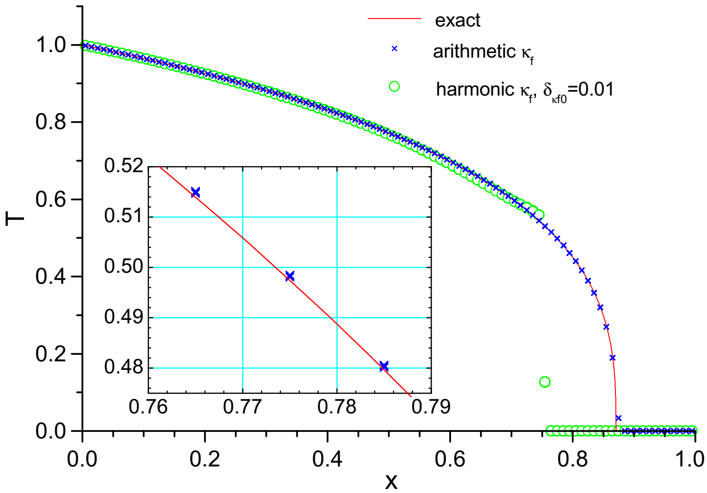

**Fig. 11.** Non-linear planar thermal wave launched into a cold wall by a fixed boundary temperature: results of simulation on a 100 × 100 square grid. Crosses: arithmetic mean $\kappa_f$; empty circles: harmonic mean $\kappa_f$.

harmonic mean formula, designed specially for strong discontinuities of $\kappa$, does not make much sense anyway. The insert in Fig. 11 shows that away from the wave front our numerical solution has typical relative errors of $|\delta T/T| \approx (2-3) \times 10^{-3}$.

### 5.6. Spherical non-linear heat wave from an instantaneous point energy source

Here, we simulate a spherically symmetric heat wave, propagating in a cold medium with a power-law conduction coefficient (63) from an instantaneous point source. We assume that at time $t = 0$ a finite amount of energy $E_0$ is instantaneously released at $r = 0$, where $r$ is the spherical radius. In this case the solution to Eq. (2) is fully analytical [11, Chapter X] and has the form

$$T(r,t) = T_c\left(1 - \frac{r^2}{r_f^2}\right)^{1/n}, \tag{66}$$

where the wave front radius is given by

$$r_f = r_f(t) = \xi_1\left(\frac{\kappa_0 t}{\rho c_V} Q_0^n\right)^{\frac{1}{3n+2}}, \tag{67}$$

and the central temperature is

$$T_c = T_c(t) = \left[\frac{n\xi_1^2}{2(3n+2)}\right]^{1/n} Q_0^{\frac{2}{3n+2}}\left(\frac{\rho c_V}{\kappa_0 t}\right)^{\frac{3}{3n+2}}. \tag{68}$$

The parameter $Q_0$ is defined as

$$Q_0 = \frac{E_0}{\rho c_V} = 4\pi\int_0^\infty Tr^2 dr, \tag{69}$$

and the dimensionless constant

$$\xi_1 = \left[\frac{3n+2}{2^{n-1}n\pi^n}\right]^{\frac{1}{3n+2}}\left[\frac{\Gamma\left(\frac{5}{2}+\frac{1}{n}\right)}{\Gamma\left(1+\frac{1}{n}\right)\Gamma\left(\frac{3}{2}\right)}\right]^{\frac{n}{3n+2}} \tag{70}$$

is obtained after we substitute Eqs. (66)–(68) into Eq. (69) and perform the integration.

For numerical simulations we select the case of $n = 2$ with $\xi_1 = 2^{7/8}\pi^{-1/2} = 1.03472826$, and set the parameter values $\rho c_V = \kappa_0 = Q_0 = 1$. One half of the unit sphere is covered by a 40 × 40 square grid of Fig. 6(a), with the $x$-axis chosen to be the cylindrical axis, and the $y$-coordinate identified with the cylindrical radius. Mirror symmetry is assumed along the $y$-axis. All the input energy $E_0$ is initially deposited in one central grid cell at $x = y = 0$. Simulation is stopped at $t = 0.3$, when the exactly calculated front radius and central temperature reach the values

$$r_f = \xi_1 t^{1/8} = 0.8901567, \quad T_c = 2^{-3/2}\xi_1 t^{-3/8} = 0.5745937. \tag{71}$$

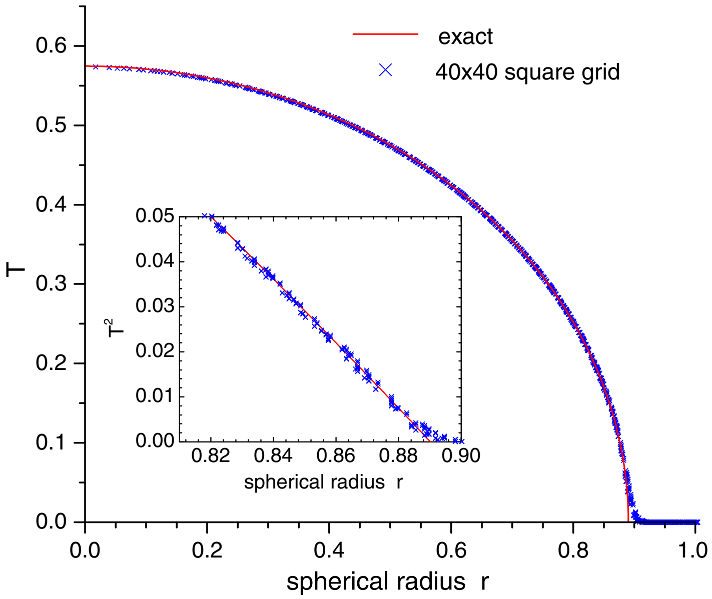

**Fig. 12.** Non-linear spherical heat wave in cold matter on a 40 × 40 square grid at $t = 0.3$.

The purpose of this test was twofold: (i) to verify how our scheme maintains the symmetry of solution (spherical) on a grid which does not possess this symmetry, and (ii) to check for possible spurious numerical effects along the axes in cylindrical $(r, z)$ coordinates. Similar to the previous problem, the time step $\Delta t$ is mainly controlled by the $\varepsilon_0$ and $T_s$ parameters. In Fig. 12 we display the results obtained with $T_s = 10^{-3}$ and $\varepsilon_0 = 0.1$, for which the numerical errors saturated on the 40 × 40 grid. Because of a very large (7 orders of magnitude) contrast between $T_s$ and the initial temperature in the central cell, this run required a relatively large number $N_{cyc} = 5558$ of time steps. In Fig. 12 all the 1600 cell-centered values of $T_{ij}$ are plotted versus spherical radii $r_{c,ij} = (x^2_{c,ij} + y^2_{c,ij})^{1/2}$ of the corresponding cell centers. One sees that all the points lie very dense on the exact solution, which means that the spherical symmetry of the solution is very well reproduced on a square grid. Also, no spurious effects are observed due to highly unequal cell volumes near and away from the rotational axes $y = 0$. Similar to the planar-wave test, the position of the wave front is quite accurately reproduced when we use the arithmetic mean $\kappa_f$ (see the insert in Fig. 12); away from the front typical relative errors of $|\delta T/T| \approx (2-3) \times 10^{-3}$ are observed.

## 6. Conclusion

We have demonstrated that a relatively simple and sufficiently accurate numerical algorithm for thermal diffusion in two dimensions can be constructed on quadrilateral grids by using the semi-implicit (SSI) approach. The algorithm is based on cell-centered temperatures and has important advantages of easy programming and computational efficiency. The proposed scheme is fully conservative and symmetric on a nine-point local stencil. It reproduces exactly linear steady-state solutions and is of the second-order spatial accuracy on not too strongly distorted quadrilateral grids (in the sense defined in Section 4.3). Smoothness of the grid turns out to be of little (if any) significance.

When tested against non-linear thermal waves, our algorithm manifests excellent accuracy on different types of not too strongly distorted quadrilateral grids, provided that a weighted arithmetic mean is used for face-centered values of the conduction coefficient. Being not fully implicit, the algorithm requires additional time step control. Our tests confirm the conclusion of Ref. [7] that in practice it is sufficient to keep (i) the relative temperature increments below 10–20%, and (ii) the relative SSI energy corrections below 1–2% in each grid cell – usual constraints dictated by the approximation accuracy considerations. We conclude that our algorithm should be well suited for 2D hydrodynamic codes with cell-centered principal variables on quadrilateral grids, and appears to be readily extendable to hexahedral grids in three dimensions.

## Acknowledgments

The authors gratefully acknowledge extensive and very fruitful discussions with P.-H. Maire and G.P. Shurtz during one day visit to UMR CELIA, CEA-CNRS-Université Bordeaux I.

## References

[1] D.S. Kershaw, J. Comp. Phys. 39 (1981) 375–395.

[2] K. Lipnikov, M. Shashkov, D. Svyatskiy, Yu. Vassilevski, J. Comp. Phys. 227 (2007) 492–512.
[3] Guangwei Yuan and Zhiqiang Sheng, J. Comp. Phys. 227 (2008) 6288–6312.
[4] J.E. Morel, J.E. Dendy Jr., M.L. Hall, S.W. White, J. Comp. Phys. 103 (1992) 286–299.
[5] M. Shashkov, S. Steinberg, J. Comp. Phys. 129 (1996) 383–405.
[6] J. Breil, P.-H. Maire, J. Comp. Phys. 224 (2007) 785–823.
[7] E. Livne, A. Glasner, J. Comp. Phys. 58 (1985) 59.
[8] R. Ramis, J. Meyer-ter-Vehn, MULTI2D – A Computer Code for Two-dimensional Radiation Hydrodynamics, Report MPQ-174, MPQ, Garching, 1992.
[9] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial Value Problems, Wiley, New York, 1967.
[10] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, Berlin, Heidelberg, 2001.
[11] Ya.B. Zel'dovich, Yu.P. Raizer, Physics of Shock-waves and High-temperature Hydrodynamic Phenomena, vol. 2, Academic Press, New York, 1967.