Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Journal of Computational Physics 203 (2005) 700–730

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An implicit compact scheme solver with application to chemically reacting flows

Mikhail Noskov *, Mitchell D. Smooke

Department of Mechanical Engineering, Yale University, New Haven, CT 06520-8284, USA

Received 3 May 2004; received in revised form 24 September 2004; accepted 24 September 2004
Available online 5 November 2004

Abstract

A novel, stable, implicit compact scheme solver that is higher order in space, suitable for modeling steady-state and time-dependent phenomena on nonuniform grids for one-dimensional configurations, is presented. Several properties of compact scheme discretizations are introduced to develop efficient algorithms for Jacobian matrix generation and Jacobian-vector multiplication using a new component form for Jacobian operations. Composite nonuniform grids are introduced that enable the implicit compact scheme solver to achieve sixth order accuracy. A robust Newton's method is employed with explicit generation of Jacobian matrices. Superior resolution characteristics of the implicit compact scheme solver are demonstrated with several steady-state and time-dependent problems for the Burgers equation. The example of the solution of stiff flame problem is given. An analysis of spectral properties of Jacobian matrices is presented, which shows that the condition number and the eigenvalue distributions behave similarly to those found in Jacobians associated with low-order discretizations. Two sparsification strategies are developed for the systematic approximation of a dense Jacobian aimed at the practical implementation of linear system preconditioning through partial Jacobians.

MSC: 65P05; 76AA05
Keywords: Implicit compact scheme solver; Newton's method; Burgers equations; Flame propagation; Spectral analysis of Jacobians

1. Introduction

Implicit numerical solvers have been used for more than half a century since their introduction by Curtiss and Hirschfelder [1] for the solution of a model stiff ordinary differential equation. By stiff, we imply here

[*] Corresponding author. Tel.: +1 203 432 4245; fax: +1 203 432 6775.
E-mail addresses: mikhail.noskov@yale.edu (M. Noskov), mitchell.smooke@yale.edu (M.D. Smooke).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.09.010

that the solution of such an equation contains several scales in time or/and length and that these scales are very disparate. Implicit techniques are used for stiff problems where explicit methods [2] encounter severe difficulties in obtaining stable results. Examples of stiff problems appear in many fields, but of particular interest to the present work are problems of chemically reacting flows. Chemically reacting flows utilizing hydrocarbon fuels occur in a variety of energy conversion processes such as combustion, propulsion, and fuel reforming, to name just a few.

Modeling flames, a typical nonlinear stiff problem, with a Newton solver is one of the application areas that frequently employs implicit time discretization [3–6]. Pseudo-time relaxation, discretized implicitly, can be used to bring the discrete vector of unknowns into the convergence domain of the steady-state problem. This latter approach has been used successfully for calculations of steady-state reacting flows in the low Mach number regime with detailed chemical kinetics and transport properties of the multicomponent mixture [7–18]. Generally, a low-order discretization in space is used in the implicit solver. The convective terms are discretized with a first order upwind scheme and the other derivative terms are discretized with centered differences (CD).

The straightforward extension of a low order in space, fully coupled, implicit solver to the case of time-dependent flames is hindered by the presence of large artificial diffusion from the first order upwind discretization of the convective terms. One way of controlling the artificial viscosity is through adaptive grid techniques that allow the grid to change at various time steps [19–21].

Alternatively, higher order spatial discretizations, without the artificial viscosity of low order methods, can be used in conjunction with an implicit solver. Several popular higher order discretizations can be found in [22–28]. A class of very accurate discretizations which can be incorporated in Newton's solver was developed by Lele [29]. The compact scheme solvers compare favorably with other existing higher order solvers [30]. For a comparative description of higher order schemes see [31].

The key idea of compact schemes, that of writing a linear algebraic equation relating the values of a function and its derivative at several grid points, is much older than the 1992 paper of Lele [29]. Two decades earlier, a third-order upwind compact scheme was introduced by Tolstykh [32], and a fourth-order accurate compact scheme was presented in the review of Orszag and Israeli [33]; that particular form is also called the Padé scheme. The merit of Lele's work is in the development of a unified approach for the generation of compact schemes for derivatives of any order with prescribed precision and stencil width. Following the unified approach, a family of boundary closures, interpolations, and numerical filters were also derived in [29]. Several extensions to compact schemes exist. Upwind compact schemes have been studied by Tolstykh and Lipavskii [34]. A coupled solution for the first and second derivatives is proposed in the method of Mahesh [35]. Gamet et al. [36] developed compact schemes for nonuniform grids.

Compact schemes have been used in modeling realistic multidimensional flows. In [37], DNS of a turbulent flame was carried out using compact schemes and an explicit time discretization. A semi-implicit solver with compact schemes was used for the calculation of complicated low Mach number 3D flows in curvilinear coordinates in [38,39]. The first implementation of compact schemes with an implicit time discretization was reported in the work of Pereira et al. [40]. They developed a Newton solver for a two-dimensional finite volume formulation of a steady-state, incompressible, constant-property flow. Within each Newton iteration, the system of linearized equations was solved with FGMRES [41] and a matrix-free approach that did not store the Jacobian entries explicitly. In [40], the linear solver contained two levels of iterations. Outer iterations were done with FGMRES, and inner iterations were done with unpreconditioned GMRES [42].

Despite its apparent attractiveness, the matrix-free method has severe drawbacks for modeling flames. The accuracy of the Gateaux derivative (within the matrix-free approach) is strongly affected by the magnitude of the perturbation parameter for the finite-digit representation of real numbers. A single perturbation parameter may be sufficiently small for the hydrodynamic variables such as velocity, yet it can be very large for some intermediate species concentrations. A large perturbation can reduce the accuracy of the matrix-vector product, which, in turn, results in an increase in the number of linear solver iterations

[31]. In addition, the balance of computational times between performing a matrix-vector product and performing two residual evaluations (required during Jacobian evaluation) may be shifted for reacting flow calculations [31].

Another word of caution should be given on using iterative linear solvers without preconditioning, such as employed in [40] for the GMRES solver within the FGMRES iterations. Such techniques can be efficient for diagonally dominant matrices. Unfortunately, the diagonal dominance is totally lost for many realistic problems. Thus, the efficiency of the unpreconditioned iterative solver is sacrificed.

In this paper, we consider the development of an implicit compact scheme solver for one-dimensional problems with particular emphasis on reacting flows. A one-dimensional configuration allows us to avoid challenges associated with multidimensional problems and it enables us to concentrate fully on the basic principles of the numerical solver. This paper is organized as follows. In the next section, the compact schemes are introduced in the case of nonuniform grids, followed by the introduction of several properties of compact schemes aimed at providing a theoretical background for the development of an implicit compact scheme solver. In Section 3, an example of the solver implementation is given, and several solutions of the Burgers equation are demonstrated. Solutions of implicit solvers (compact scheme and traditional low order) are compared, and the accuracy of the implicit compact scheme solver is analyzed. The ability of the new solver to handle stiff problems is demonstrated by modeling the propagation of an unsteady flame. In addition, the spectral properties of compact scheme Jacobians are analyzed. Two strategies for Jacobian approximation are investigated. Section 4 concludes this paper with a summary of the work presented.

2. Compact scheme properties

In the present work, we employ the compact schemes of [36] with built-in nonuniform grid metrics. Following [29,36], the calculation of the second derivative is carried out using the corresponding compact scheme, instead of applying the first derivative operator twice. Defined on a 1D nonuniform grid $\{x_i, i = 1, \ldots, N\}$, a function $u(x)$ is smooth, i.e., it is continuous, together with all of its derivatives, on the domain $x_1 \leqslant x \leqslant x_N$. The compact scheme discretizations of the first ($u' = u_x$) and second ($u'' = u_{xx}$) derivatives are written in matrix form as

$$A_x u' = B_x u, \tag{1}$$

$$A_{xx} u'' = B_{xx} u. \tag{2}$$

Here, A_x, B_x, A_{xx}, and B_{xx} are banded matrices, constant for a given grid. The expressions for the evaluation of entries within these matrices are defined by matching the terms of Taylor series expansions [29,36].

The development of compact schemes on nonuniform grids contains one delicate difficulty not present in the uniform grid case. If very high order is sought for the discretization on nodes in a selected stencil, then the discretization error may grow without bound for a certain combination of grid step sizes. Unfortunately, this problem is not explained in [36]. The reason for this problem is as follows. The truncation error term contains numerous algebraic expressions involving grid step sizes. There could be a division of some term by an algebraic expression that vanishes for some particular combination of grid steps, leading to an unstable discretization scheme. However, the same scheme can be stable for all other combinations of grid steps. Unbounded growth of the truncation error, which may occur for both first and second derivatives, can be avoided if a compact scheme is generated that has no subtraction in any denominator within the expression for the local truncation error. This goal is accomplished by using a reduced order of accuracy on nonuniform grids.

We use the same compact scheme discretization as derived in [36] where possible. In addition, we use the following relationships for the second derivative at the boundary nodes:

$$u_1'' + 11u_2'' = \beta_{11}^2 u_1 + \beta_{12}^2 u_2 + \beta_{13}^2 u_3 + \beta_{14}^2 u_4 + \mathrm{LTE}_1^2, \tag{3}$$

$$10u_1'' + u_2'' + 10u_3'' = \beta_{21}^2 u_1 + \beta_{22}^2 u_2 + \beta_{23}^2 u_3 + \beta_{24}^2 u_4 + \mathrm{LTE}_2^2, \tag{4}$$

where the coefficients for nonuniform grids are defined as

$$\begin{aligned} \beta_{11}^2 &= \frac{6h_2 + 48h_3 + 24h_4}{h_2(h_2+h_3)(h_2+h_3+h_4)}, \qquad \beta_{12}^2 = \frac{18h_2 - 48h_3 - 24h_4}{h_2h_3(h_3+h_4)}, \\ \beta_{13}^2 &= \frac{24h_3 - 18h_2 + 24h_4}{h_3(h_2+h_3)h_4}, \qquad \beta_{14}^2 = \frac{18h_2 - 24h_3}{h_4(h_3+h_4)(h_2+h_3+h_4)}, \end{aligned} \tag{5}$$

and

$$\beta_{21}^2 = \frac{3h_2 + 21h_3 + 12h_4}{5h_2(h_2+h_3)(h_2+h_3+h_4)}, \qquad \beta_{22}^2 = \frac{9h_2 - 21h_3 - 12h_4}{5h_2h_3(h_3+h_4)}, \tag{6}$$

$$\beta_{23}^2 = \frac{9h_3 - 9h_2 + 12h_4}{5h_3(h_2+h_3)h_4}, \qquad \beta_{24}^2 = \frac{9h_2 - 9h_3}{5h_4(h_3+h_4)(h_2+h_3+h_4)}. \tag{7}$$

The expression for node 1 is given by Eq. (3), and the expression for node 2 is given by Eq. (4). The grid step is $h_i = x_i - x_{i-1}$. Expressions for second derivatives on nodes $N-1$ and N can be obtained in a similar way; see also [29,36]. A summary of the formal orders of accuracy for the compact schemes used in the present work is given in Table 1.

2.1. Computational stencils of compact schemes

Each of the compact scheme equations given in Table 1 uses a fixed number of nodes to define relationships for the derivatives. For example, there are four nodes employed in Eqs. (3) and (4). In the literature, the set of nodes used in these linear equations is sometimes called a stencil. In the discretization scheme presented above, the maximum number of nodes per equation is five.

Traditionally, stencil width is defined as the number of nodes needed to generate the approximation to the derivative at each grid node. Multiplying Eq. (1) on the left by A_x^{-1}, we get

$$u' = (A_x^{-1}B_x)u \equiv C_x u. \tag{8}$$

Matrix C_x, of size $N \times N$, is dense, and its entries define numerical coefficients of the discretization stencils for each node. Therefore, matrix C_x will be called a stencil matrix. In a similar manner, another dense $N \times N$ stencil matrix C_{xx} can be obtained for the second derivative. Dense matrices C_x and C_{xx} imply that the stencil width in the traditional sense is N for first and second derivatives. In other words, the value of a derivative at a given node depends on the function values at all grid nodes in the direction in which the derivative is being taken.

Table 1
Orders of accuracy for the compact scheme discretizations (last two columns)

Derivative	Node type	Uniform grid	Nonuniform grid
First	Inner, $3 \leqslant i \leqslant N-2$	6	4
First	Boundary, $i = 1, N$	3	3
First	Boundary, $i = 2, N-1$	4	4
Second	Inner, $3 \leqslant i \leqslant N-2$	6	3
Second	Boundary, $i = 1, N$	2	2
Second	Boundary, $i = 2, N-1$	2	2

When tensor product grids are used for two-dimensional problems, stencil matrices can be defined for each coordinate direction. Moreover, cross-derivatives appear in the full Navier–Stokes equations, and the discretization stencil for the cross-derivative u_{xy} contains all nodes in the two-dimensional computational domain. Thus, unlike the low order discretization methods that use a small, nine-point stencil in 2D, the compact scheme discretization produces very large stencils.

2.2. Linearity of compact schemes

For simplicity, we consider a nonlinear residual function $f(u,u')$ of variables $u(x)$ and $u'(x)$, where x is a spatial variable. The corresponding nonlinear governing equation is $f(u,u') = 0$. The spatial derivative u' is discretized with compact schemes. Newton's method can be used to minimize the residual f, with the Jacobian matrix calculated explicitly as follows. We define the Jacobian matrix entry $J(f,u)_{ij}$ as a one-sided derivative of the residual f at node i, taken with respect to the function u at node j, i.e.,

$$J(f,u)_{ij} = \lim_{\delta u_j \to 0} \frac{f_i(u+\delta u) - f_i(u)}{\delta u_j}, \tag{9}$$

where $u = \{u_1, \ldots, u_j, \ldots, u_N\}^{\mathrm{T}}$ and $\delta u = \{0, \ldots, 0, \delta u_j, 0, \ldots, 0\}^{\mathrm{T}}$. If $i \neq j$, then the perturbed value $u_j + \delta u_j$ results in a perturbed value of the residual only through the perturbed spatial derivative $\widetilde{u_i'}$, where $\widetilde{(\cdot)}$ denotes a perturbed value. Similar to Eq. (8), the expression for the perturbed compact derivative can be rewritten as

$$\widetilde{u_i'} = \sum_{k=1}^{N} (C_x)_{ik} \widetilde{u_k} = (C_x)_{ij} \widetilde{u_j} + \sum_{\substack{k=1 \\ k\neq j}}^{N} (C_x)_{ik} u_k = u_i' + (C_x)_{ij} \delta u_j. \tag{10}$$

The expression

$$\widetilde{u_i'} = u_i' + (C_x)_{ij} \delta u_j \tag{11}$$

is exact. Here and further in the text, no summation is implied for a repeated index. Eq. (11) defines *the property of linearity of compact schemes*, or that a compact discretization is a linear function of nodal entries. Similarly, the property of linearity can be established for other spatial derivatives.

Using the linearity of compact schemes, each Jacobian entry $J(f,u)_{ij}$ can be computed in a straightforward way:

$$J(f,u)_{ij} = (f_u)_i \delta_{ij} + (f_{u'})_i (C_x)_{ij}. \tag{12}$$

Here, $f_u = \partial f / \partial u$ and $f_{u'} = \partial f / \partial u'$ are the partial derivatives of residual f with respect to u and u', and δ_{ij} is the Kronecker delta. Analytical expressions for these partial derivatives can be obtained and evaluated at each node.

2.3. Jacobian matrix-vector product

Using Eq. (12), a Jacobian matrix-vector product $w = Jv$ is

$$w_i = \sum_{j=1}^{N} J_{ij} v_j \tag{13a}$$

$$= ((f_u)_i + (f_{u'})_i (C_x)_{ii}) v_i + \sum_{\substack{j=1 \\ j\neq i}}^{N} (f_{u'})_i (C_x)_{ij} v_j = (f_u)_i v_i + (f_{u'})_i \sum_{j=1}^{N} (C_x)_{ij} v_j. \tag{13b}$$

The last term on the second line can be simplified if one recalls that the dot product of the stencil matrix with a vector produces the derivative of that vector (viz. Eq. (8)). Therefore, the following exact expression is obtained for the Jacobian-vector product:

$$w_i = (f_u)_i v_i + (f_{u'})_i v'_i. \tag{14}$$

The first term in Eq. (14) reflects the presence of terms local to node i in the residual f, such as a time derivative or the source terms due to a chemical reaction. The second term in Eq. (14) contains information about the spatial derivatives. Expressions f_u and $f_{u'}$ will be called the components of the Jacobian, and the Jacobian-vector product, as calculated by Eq. (14), will be called the component form of Jacobian-vector product. Unlike the matrix-free approach, the component product is exact.

The matrix-vector product given in Eq. (13a), which we will refer to as a direct product, requires $O(N^2)$ floating point operations to obtain vector w. Another unfortunate aspect of the direct product is that all Jacobian entries J_{ij} must be stored, consuming $O(N^2)$ double precision memory locations. The use of the component form of the Jacobian-vector product provides a computationally cheaper alternative. The first term of Eq. (14) is evaluated similar to the direct matrix-vector product, except that it requires $O(N)$ floating point operations to obtain vector $w_1 = f_u v$. The second term in Eq. (14) is formed in two steps. First, the spatial derivative is evaluated for vector v using the compact scheme, as in Eq. (1), with a computational cost of $O(8N)$ operations. The second step is to calculate the outer vector product of the derivative vector v' with the vector of coefficients $f_{u'}$, involving $O(N)$ operations. The storage for the coefficient vectors f_u and $f_{u'}$ is $O(2N)$. In total, the component form of the Jacobian-vector product requires only $O(10N)$ floating point operations and the storage of $O(5N)$ double precision numbers.

In multidimensional chemically reacting flows, additional types of derivatives are used, and the number of unknowns per node is large (from 20 to upwards of 200 unknowns). However, the comparison based on memory consumption and operation counts always favors the component product over the direct product. The difference is usually of several orders of magnitude, both in memory allocation and in computational cost [31].

2.4. Determination of partial Jacobians

One of the common features of compact scheme stencils is the rapid exponential decay in the amplitude of the nodal coefficients with distance from the central node. It can be shown that this decay of stencil matrix entries stems from the exponential decay of the off-diagonal entries of the inverse of banded matrix A_x [43]. Due to the rapid decay in the entries on each row of C_x and C_{xx}, only few nodes to the left and to the right of the central node have coefficients with a magnitude larger than some prescribed threshold value. These nodes constitute a partial stencil, including the central node. Several sparsification strategies with threshold θ can be devised to define partial stencils and, therefore, the structure of a partial Jacobian. For each grid node i, we can introduce a subset of nodes representing a partial stencil $\Phi_i(j)$ for evaluation of spatial derivatives, where $1 \leqslant i,j \leqslant N$. Three sparsification strategies are given below:

$$k \in \Phi_i(j) \quad \text{if } |(C_x)_{ik}| > \theta \max_m |(C_x)_{im}|, \tag{15}$$

$$k \in \Phi_i(j) \quad \text{if } |(C_{xx})_{ik}| > \theta \max_m |(C_{xx})_{im}|, \tag{16}$$

$$k \in \Phi_i(j) \quad \text{if } |(J)_{ik}| > \theta \max_m |(J)_{im}|, \tag{17}$$

where typical values of θ might be in the range of 10^{-4} to 10^{-2}. In these sparsification strategies, we always include all grid nodes between the leftmost and rightmost nodes of the partial stencil given by the subset $\Phi_i(j)$. The Jacobian formed with the partial stencils of the compact scheme will be called a partial Jacobian J_p. The partial Jacobian approximation fits the operation of a preconditioner solve naturally, because this operation delivers an approximate solution to the linear problem.

2.5. Stability analysis of the implicit compact scheme discretizations

The stability of compact scheme solvers has been analyzed in the literature for explicit time discretizations and semi-discrete problems. Results of the semi-discrete form of the stability analysis are applicable to both explicit and implicit time discretizations. In [29], it was found that the explicit Euler method is unconditionally unstable for the advection equation discretized with compact schemes. For the same advection equation, conditional stability was established using Runge–Kutta methods for the time discretizations. The explicit Euler and Runge–Kutta discretizations of the model conduction equation were both found to be conditionally stable.

Carpenter et al. [44], who studied the time stability of the semi-discrete form of the model advection equations on uniform grids using G–K–S theory, showed that some but not all combinations of inner schemes and boundary closures given by Lele [29] are time-stable. One of the time-stable schemes denoted in [44] as 3–4–6–4–3 (viz. Table 1 for the first derivative) and originally developed in [29] for uniform grids has found its use in the work of Gamet et al. [36].

In [36], a von Neumann (Fourier) stability analysis was carried out for semi-discrete forms of the advection and conduction equations on nonuniform grids, both of which were found to be stable. Next, the effect of compact scheme boundary closures on the stability of differential operators was investigated using a matrix stability analysis. For an explicit, third-order Runge–Kutta time discretization, it was shown that the compact scheme developed is conditionally stable for a range of CFL numbers CFL = $c\Delta t/h_{\min}$ < $\mathrm{CFL}_{\max}$ for the advection equation and for a range of Fourier numbers $Fo = \mu\Delta t/h_{\min}^2 < Fo_{\max}$ for the conduction equation.

In the present work, the implicit Euler method is to be used for nonuniform grids, together with the compact scheme of Gamet et al. [36] and modifications given by Eqs. (3) and (4). The matrix stability analysis is given below for the model one-dimensional advection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{18}$$

and the model one-dimensional conduction equation

$$\frac{\partial u}{\partial t} - \mu\frac{\partial^2 u}{\partial x^2} = 0. \tag{19}$$

Here, $u(x,t)$ is a function of space and time, and c and μ are constants that represent advection velocity and kinematic viscosity, respectively.

The matrix stability analysis developed by Lax and Richtmyer [45] is applicable to fully discrete problems with imbedded boundary conditions. In fully discrete form (or stationary iterative form), the time-stepping (or iterative) procedure for a solution vector u^{n+1} at time level (or iteration) $n + 1$ is written

$$u^{n+1} = Ku^n + p, \tag{20}$$

where an iteration matrix K and an update vector p describe spatial operators and source terms given by a differential equation. A necessary condition for the stability of a linear problem, such as given by Eqs. (18) and (19), is that the spectral radius of the iteration matrix is bounded as

$$\rho(K) \leqslant 1. \tag{21}$$

In [45], a sufficient condition for the stability of a linear problem with constant coefficients is given by the inequality $\|K\| \leqslant 1$. Unfortunately, the sufficient condition may be too strong for some practical methods and may be inadequate for the case of systems of nonlinear equations.

The discrete form of Eq. (18), in the case of the first order implicit Euler method, is

$$u^{n+1} = (I + c\Delta t C_x)^{-1} u^n = K_c u^n, \tag{22}$$

and the discrete form of Eq. (19) is

$$u^{n+1} = (I - \mu\Delta t C_{xx})^{-1} u^n = K_\mu u^n. \tag{23}$$

Here, u^{n+1} and u^n represent solution vectors at time levels $n+1$ and n; Δt – the time step of the Euler method; I – the identity matrix; C_x and C_{xx} – the stencil matrices for first and second derivatives, respectively; and K_c and K_μ – the iteration matrices. To accommodate the boundary condition for Eq. (18), the first row of the stencil matrix is set to zero except for the main diagonal entry, which is unity. To implement the boundary conditions for Eq. (19), the first and the last rows of the stencil matrix are set to zero except for the main diagonal entries, which are set to unity. The effect of the boundary closures of the compact scheme is nonetheless included in the iteration matrix through coefficients within other rows of the stencil matrix. For the discretization to be stable, it is necessary that all eigenvalues of K_c and K_μ lie within a unit circle, as given by Eq. (21) [29,36,45].

The eigenvalue analysis of iteration matrices of Eqs. (22) and (23) in the analytical form for the general case of N grid points does not seem to be possible due to numerous entries of dense stencil matrices C_x

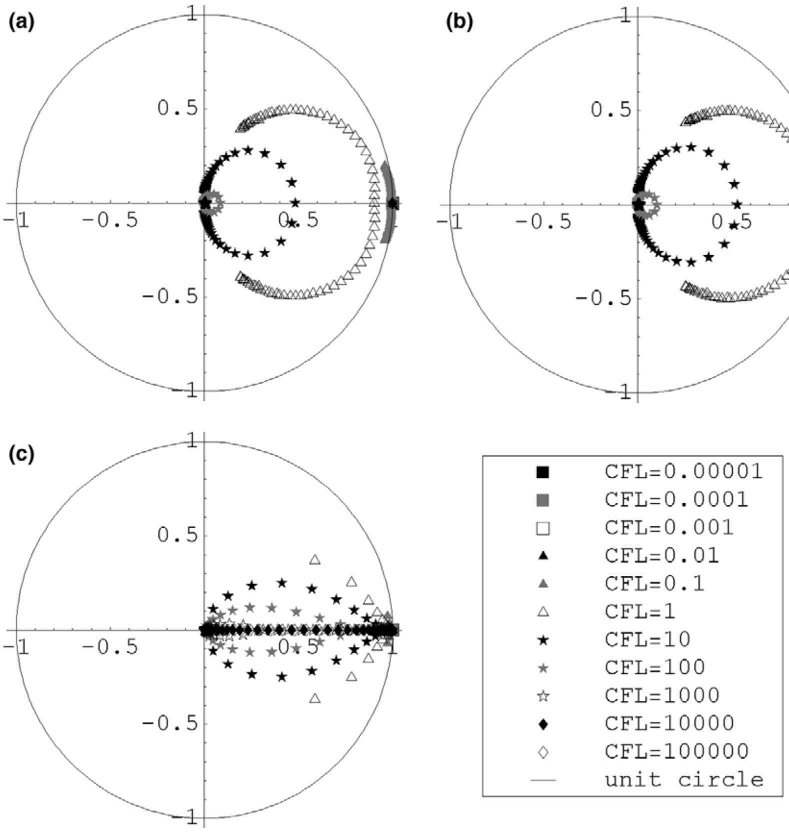

Fig. 1. Distribution of the eigenvalues of the iteration matrix for the advection equation on the complex plane for selected CFL numbers. Number of nodes: 80. Horizontal coordinate: real part. Vertical coordinate: imaginary part. Panel (a) corresponds to uniform grid, panel (b) – random grid, and panel (c) – geometric grid.

and C_{xx}. Instead the eigenvalue analysis is carried out for several practical cases following [36]. Three grids are selected: a uniform grid; a randomly spaced grid; and a grid with smoothly varying spacing. Each grid has 80 nodes. The computational domain is $0 \leqslant x \leqslant 5$. As in [29,36], the boundary conditions are homogeneous for the advection equation: $u(0,t) = 0$; and for the conduction equation: $u(0,t) = u(5,t) = 0$. The uniform grid has $h_i \simeq 0.06329$; the random grid has a randomly varying step size $0.052638 \leqslant h_i \leqslant 0.0732$; and the geometric grid has a constant ratio of decreasing step size $h_{i+1} = h_i/1.3$.

Shown in Figs. 1 and 2 are the distributions of eigenvalues of K_c and K_μ, respectively. As can be seen from Fig. 1, for all cases studied, the eigenvalues are located inside the unit circle on the complex plane. Note that the CFL number is calculated based on the smallest grid step size. It is expected that for even larger values of time step Δt and, hence, the CFL number, the condition given by Eq. (21) will remain satisfied. All eigenvalues of K_μ are real-valued and positive. As can be seen from Fig. 2, in all cases considered, the condition $\rho(K_\mu) \leqslant 1$ is satisfied. The *Fo* number is calculated based on the smallest grid step size. It is again expected that for even larger values of time step Δt and, hence, the *Fo* number, the condition given by Eq. (21) will remain satisfied.

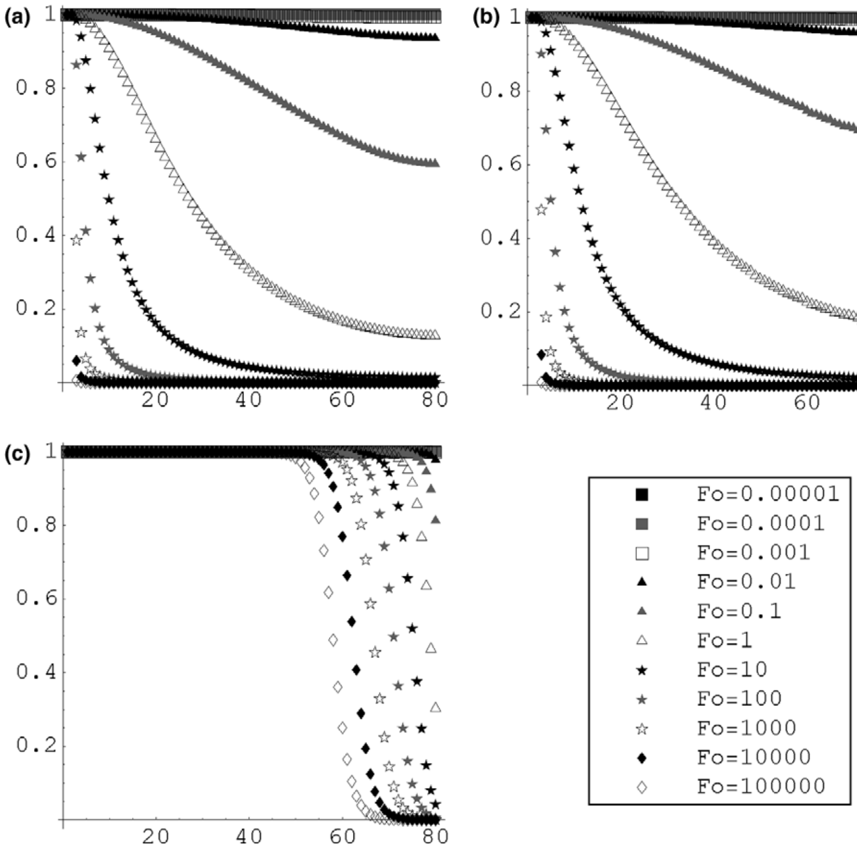

Fig. 2. Distribution of the eigenvalues of the iteration matrix for the conduction equation for selected Fourier numbers. Number of nodes: 80. Horizontal coordinate: index of eigenvalue. Vertical coordinate: eigenvalue (real, in the decreasing order of size). Panel (a) corresponds to uniform grid, panel (b) – random grid, and panel (c) – geometric grid.

3. Computational results

The theory of the previous section provides sufficient guidance for the development of an implicit solver for a one-dimensional (1D) configuration. Consider the following equation for the function $u(x,t)$:

$$f = \frac{\partial u}{\partial t} + w\frac{\partial u}{\partial x} - \mu\frac{\partial^2 u}{\partial x^2} + g(u,x,t) = 0. \tag{24}$$

Here, w represents the convection velocity; μ – the viscosity; and $g(u,x,t)$ – the source term. If $w = u$, then this is the nonlinear Burgers equation. If $w = c =$ constant, then the linear Burgers equation is obtained. A detailed list of possible analytical solutions is given by Benton and Platzman [46]. Some frequently used solutions of the steady-state and time-dependent Burgers equations are given in [24].

In the current work, the solution of the steady-state Burgers equation proceeds in the following manner. Starting from some initial guess, the solution vector is advanced in time using the noniterative time-linearization of Briley and McDonald [47,48]. The effect of the choice of the initial guess is minimal as long as the initial time step is sufficiently small ($\Delta t^0 = 10^{-3}$). The time step size is increased gradually (i.e., $\Delta t^{n+1} = 1.4\Delta t^n$) so that the effect of the transient term becomes smaller during time-stepping. Once some specified value of the time step is reached ($\Delta t^n = 10$), Newton's method for the steady-state problem is engaged [49–51], using the most recent iterate obtained in the process of time-relaxation as the starting guess. The basic form of Newton's method is

$$J(f^k, u^{n+1,k})\Delta u^{n+1,k} = -f^k, \tag{25}$$

where k indicates the iteration number within Newton's method. Here, $J(f^k,u^{n+1,k})$ represents the Jacobian matrix for the residual vector $f^k = f(u^{n+1,k},u^n)$ with respect to the vector $u^{n+1,k}$; $\Delta u^{n+1,k}$ – the increment vector; and $u^{n+1,k} = u^{n+1,k-1} + \Delta u^{n+1,k-1}$ – the solution vector.

The solution of the time-dependent Burgers equation has many features in common with the process of time-relaxation of the steady-state problem, but now the time step size is controlled by the CFL and Fo numbers, i.e., $\Delta t = \min\left(\mathrm{CFL}\frac{h_{\min}}{w}, Fo\frac{h_{\min}^2}{\mu}\right)$. The solution must be accurate at all time levels. For this purpose, the following restriction is set: CFL = Fo = 0.1. Stable calculations of the time-dependent Burgers equation can be carried out with much larger time steps, but then solution accuracy will be sacrificed. Three implicit methods are employed: the first-order Euler method, the second-order Crank–Nicolson method (also involving two time levels) [24], and second-order backward differences (involving three time levels). At each time level, the equation is solved with the Briley–McDonald approach for the Euler and Crank–Nicolson discretization and with the Newton solver for the three-time-level backward discretization. Generally, only first order temporal accuracy can be obtained for nonlinear equations with the Briley–McDonald method. However, for linear equations, the time linearization produces second-order-accurate results if a two-level Crank–Nicolson time discretization is used.

Because the grid is held fixed, compact scheme matrices A_x, B_x, C_x, A_{xx}, B_{xx}, and C_{xx} need to be evaluated only once at the beginning of the calculations. The stencil matrices C_x and C_{xx} are dense, and each component must be stored; for example, one stencil matrix, with double precision representation of its entries, requires 80 kilobytes of memory for a 100-node grid.

During the time-stepping procedure, the following operations are carried out routinely. Tridiagonal systems (Eqs. (1) and (2)) are solved with the highly efficient direct Thomas algorithm [24]. According to Eq. (12), the Jacobian elements for the nonlinear time-dependent Burgers equation are given in component form as

$$J(f,u)_{ij} = (f_u)_i\delta_{ij} + w_i(C_x)_{ij} - \mu(C_{xx})_{ij}, \tag{26}$$

$$(f_u)_i = \frac{1}{\Delta t} + (w_u)_i(u_x)_i + (g_u)_i. \tag{27}$$

The derivatives w_u and g_u are formed analytically for the particular Burgers equation, e.g., $w_u = 1$ for the nonlinear Burgers equation. The modeling of one-dimensional problems renders a significant simplification with respect to the linear system solver for Eq. (25). Because of the small number of unknowns, the full Jacobian matrix can be stored, and a direct solver with Gaussian elimination can be used [52].

For comparison purposes, calculations with traditional (low order) finite difference discretizations have also been carried out, in which first derivatives in space are discretized using first order upwinding and second derivatives are discretized using centered differences (of second order accuracy on uniform grids). On nonuniform grids, both derivative discretizations are first-order accurate. The computational stencil for the low order scheme has three nodes. The low order discretization code has been obtained by replacing the compact scheme discretization with the low order discretization and by replacing the compact scheme stencil matrices with the stencil matrices of the low order scheme. The linear system with the low order Jacobian J_{low} is solved with the efficient Thomas algorithm.

3.1. Steady-state model problem without source terms

From [24], the steady-state form of the nonlinear Burgers equation (Eq. (24)) without source terms has the analytical solution

$$u(x) = u_0\eta\left(1-\exp\left(\eta R\left(\frac{x}{L}-1\right)\right)\right)\left(1+\exp\left(\eta R\left(\frac{x}{L}-1\right)\right)\right)^{-1} \tag{28}$$

for the boundary conditions $u(0) = u_0$ and $u(L) = 0$. The parameter R has the properties of the Reynolds number and is defined as $R = \frac{u_0 L}{\mu}$. The parameter η is the root of the nonlinear equation

$$\frac{\eta-1}{\eta+1} = \exp(-\eta R). \tag{29}$$

The following set of numerical parameters is selected: $L = 5$; $u_0 = 1$; and $\mu = 0.5$. For these settings, we obtain $R = 10$ and $\eta \approx 1.00009072$. The presence of the exponential function in the solution results in significant gradients of the function $u(x)$, together with its derivatives, near $x = L$. Therefore, more grid nodes should be clustered in this region. For a given weight function $W(x)$, the grid nodes are equidistributed according to the expression $W(x_{i+1}) - W(x_i) = \Delta W =$ constant, and the total number of nodes is given by $N = 1 + (W(L) - W(0))/\Delta W$. Here, a family of nonuniform grids has been generated with a weight function $W(x)$ based on the arclengths of $u(x)$ and its derivatives $u^{(\mathrm{IV})}(x)$ and $u^{(\mathrm{V})}(x)$:

$$W(x) = \int_0^x (1+u(t)^2)\mathrm{d}t + \psi_1\int_0^x (1+u^{(IV)}(t)^2)\mathrm{d}t + \psi_2\int_0^x (1+u^{(V)}(t)^2)\mathrm{d}t. \tag{30}$$

In selecting this weight function, a balance is sought to obtain an efficient refinement strategy suitable for both low and high order discretizations. The choice of the even and odd derivatives ($u^{(\mathrm{IV})}(x)$ and $u^{(\mathrm{V})}(x)$) for the calculation of $W(x)$ is intended to provide sufficient resolution characteristics for even and odd modes present within the solution. No claim is made here that this weight function produces an optimal grid for both the compact scheme and the low order methods for the given problem. However, this weight function results in smoothly distributed grid nodes that are clustered at the steep gradients. Some test calculations have been carried out with other forms of weight functions that also produce grid refinement near the steep gradients. No radical changes in the comparative characteristics of the compact scheme vs. the low order scheme have been noticed. Some arbitrariness in the selection of derivative orders and the choice of constants (here, $\psi_1 = 1/3$ and $\psi_2 = 1/40$) provide an offset to the use of the exact solution $u(x)$ within the weight function. In practical calculations, the solution is not known, and the grid is refined based on the presence of steep gradients.

A summary of the results is given in Table 2. One can see the superior accuracy of the compact scheme solutions in comparison to the solutions obtained by the traditional low order schemes. With only a 10-node grid, the compact scheme method generates a solution with about 1% relative error (viz. Table 2). In order to obtain a similar level of accuracy, the low order scheme requires about 380 nodes, or a 38-fold increase in the size of the grid. The smaller number of grid nodes in the case of compact schemes is offset by the need to use the more expensive Gaussian elimination of a dense Jacobian. We investigate the operation count in detail in Section 3.3 for a time-dependent problem.

The analytical solution for the nonlinear Burgers equation (Eq. (28)), together with the above mentioned boundary conditions, is shown in Fig. 3, along with the compact scheme solution on the 10-node grid and the low order scheme solutions for two grids with 10 and 160 nodes. As seen in Fig. 3, the solution of the compact scheme on the 10-node grid visibly coincides with the analytical solution, while the low order solution on the 160-node grid still deviates from the analytical profile.

It is instructive to introduce another solution: that of the steady-state nonlinear Burgers equation with the same boundary conditions but with an artificially augmented viscosity

$$\mu_{\text{eff}} = \mu + \frac{\widehat{u}\widehat{h}}{2}. \tag{31}$$

The second term in Eq. (31) is the artificial viscosity due to the low order discretization of the convective term. Here, $\widehat{u}$ and $\widehat{h}$ are some characteristic values for $u(x)$ and the grid step size h, respectively. The average values are used, viz.

$$\widehat{u} = \frac{1}{L}\int_0^L u(t)\mathrm{d}t, \qquad \widehat{h} = \frac{1}{N-1}\sum_{i=2}^{N} h_i. \tag{32}$$

The analytical solution of the Burgers equation with augmented viscosity (Eq. (31)) is calculated for $\widehat{u}$ and $\widehat{h}$ of the 10-node grid (viz. Fig. 3, curve "b"). The solution of the low order scheme on the 10-node grid almost coincides with curve "b". Thus, the low order scheme, in effect, solves the Burgers equation with augmented viscosity, given by Eq. (31). For the low order scheme, the original Burgers equation solution can be recovered only by reducing the grid step size h and thus reducing the artificial viscosity. The linear dependence of the artificial viscosity on the grid step size makes the process of reducing the artificial viscosity expensive. A similar analysis of modified (augmented) equations in which the discretization error terms are explicitly introduced in the differential equations can be found in [53].

The compact scheme developed here contains relationships of different orders of accuracy for first and second derivatives and for inner and boundary nodes (viz. Table 1). A summary of calculations for uniform and nonuniform grids is presented in Table 3. Two different approaches are selected for nonuniform grids. In the first approach (geometric grid), the refinement of the grid is obtained by reducing the grid step size in

Table 2
Model problem without source terms

Run	Discretization	Nodes	$h_{\min}$	$h_{\max}$	Absolute error	Relative error
1	Compact	10	0.13726	0.869274	0.00545144	0.00995833
2	Compact	160	0.0076758	0.0517357	1.60552×10^{-7}	2.9014×10^{-7}
3	Low order	10	0.13726	0.869274	0.154091	0.275038
4	Low order	160	0.0076758	0.0517357	0.0130239	0.0237304
5	Low order	320	0.00382583	0.0258105	0.00657206	0.0120105
6	Low order	380	0.00322014	0.0217259	0.00554221	0.0101335

Summary of the results for calculations with the compact scheme and the low order scheme on the nonuniform grids generated with weight function $W(x)$.

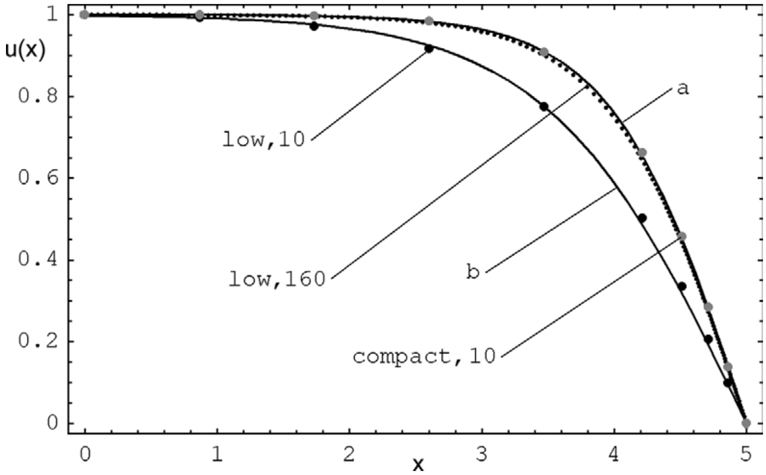

Fig. 3. Steady-state Burgers equation solutions on the nonuniform grids generated with the weight function $W(x)$ with 10 and 160 nodes. The compact scheme solution is denoted by gray dots. The low order solution is given by black dots. Curve "a" – analytical solution of the nonlinear Burgers equation. Curve "b" – analytical solution of the nonlinear Burgers equation with augmented viscosity.

the direction toward the steep gradients, i.e., $h_{i+1} = h_i/1.1$. The second approach introduces a random generation of node locations, based on a uniform statistical distribution; at the same time, the step size h_i is restricted between some minimum and maximum values to avoid drastic changes in step size between two neighboring grid intervals.

The overall order of accuracy is calculated based on the maximum grid step size $h_{\max,m}$ in the domain and the maximum relative error of the solution ε_m for each set of grids. The best linear fit $(a + bx)$ is found for decimal logarithms of step size and relative error $\{\log(h_{\max,m}),\log(\varepsilon_m)\}$ for each set using a least squares

Table 3
Model problem without source terms

Run	Grid	Nodes	$h_{\max}$	$h_{\min}$	Absolute error	Relative error
1	Uniform	10	0.555556	0.555556	0.0160671	0.0318314
2	Uniform	20	0.263158	0.263158	0.00100837	0.00391916
3	Uniform	30	0.172414	0.172414	0.000141092	0.000826275
4	Uniform	40	0.128205	0.128205	3.36188×10^{-5}	0.000263614
5	Uniform	80	0.0632911	0.0632911	1.0285×10^{-6}	1.6269×10^{-5}
6	Geometric	10	0.789275	0.368203	0.00351177	0.00996305
7	Geometric	20	0.543395	0.0977343	6.83564×10^{-5}	7.17214×10^{-5}
8	Geometric	30	0.485128	0.0336404	3.98545×10^{-5}	5.36588×10^{-5}
9	Geometric	40	0.465868	0.0124549	3.28249×10^{-5}	4.80173×10^{-5}
10	Geometric	80	0.45479	0.000268646	2.92156×10^{-5}	4.44518×10^{-5}
11	Random	10	0.653411	0.471848	0.00351177	0.173639
12	Random	20	0.311007	0.230678	6.83564×10^{-5}	0.00428565
13	Random	30	0.196744	0.144396	3.98545×10^{-5}	0.000205675
14	Random	40	0.149511	0.108282	3.28249×10^{-5}	0.000224959
15	Random	80	0.0732001	0.0526377	2.92156×10^{-5}	1.94477×10^{-5}

Summary of the results for calculations with the compact scheme on uniform and nonuniform grids.

method; these fits are plotted in Fig. 4. The slope b of the linear fit provides the overall order of accuracy of the numerical scheme.

The numerical solutions using the compact scheme method on the uniform grids exhibit approximately fourth order accuracy ($b = 3.51$). The solutions using the same method on the nonuniform random grids have again a fourth order accuracy ($b = 4.19$). The solutions on the nonuniform geometric grids show a more complicated character. Therefore, two linear fits are given for this grid type. The first fit, based on all data points for the geometric grids, indicates a tenth order accuracy ($b = 10.1$). The second fit, taken with the data of the 10-node solution removed, shows only a third order accuracy ($b = 2.66$). The error decays rapidly between the solutions on the 10-node grid and the 20-node grid, because the refinement achieved with the geometric grids is sufficient to remove a dominant mode in the solution error. However, the subsequent addition of more grid nodes does not lead to such a drastic gain of accuracy, since more modes of numerical error have to be removed. It is worth pointing out that for the finest grids with 80 nodes, the best accuracy is obtained for the uniform grid.

3.2. Steady-state solution for a bell-shaped profile

In flames, many intermediate chemical species are distributed with sharp bell-shaped profiles, because they are present inside the flame and quickly disappear away from the flame. Accurately solving the intermediate species is important for flame modeling. A Burgers equation whose solution is similar to that of the intermediate species is now modeled. Consider the steady-state nonlinear ($w = u$) Burgers equation (Eq. (24)) with the source term given by

$$g(x) = \frac{40}{e^{20(x-2)^2}} - \frac{158}{e^{10(x-2)^2}} - \frac{20x}{e^{20(x-2)^2}} + \frac{160x}{e^{10(x-2)^2}} - \frac{40x^2}{e^{10(x-2)^2}}. \tag{33}$$

The viscosity coefficient is set to $\mu = 0.1$, the boundary conditions are $u(0) = u(L) = 0$, and the dimension of the domain is $L = 5$. The analytical solution to this problem is a function with a bell-shaped profile given by the expression

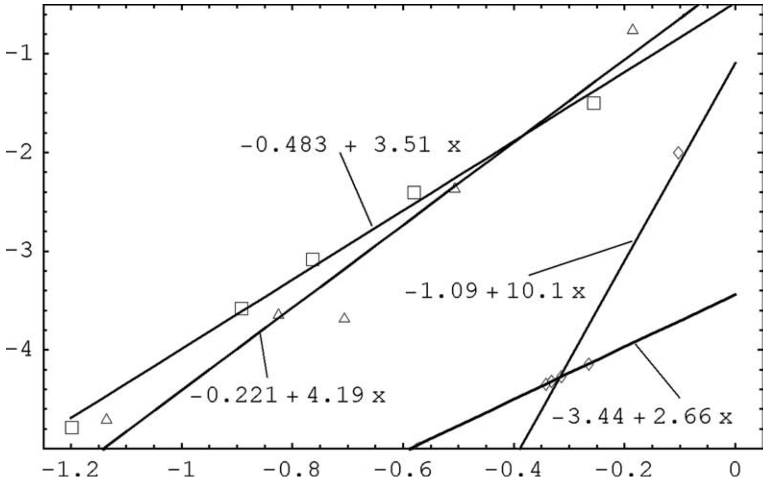

Fig. 4. Decimal logarithm of the maximum relative error in the compact scheme solution of the steady-state Burgers equation vs. decimal logarithm of the maximum grid step size. Solution data is represented with symbols. Boxes – uniform grids with 10, 20, 30, 40, 80 nodes; triangles – nonuniform random grids with 10, 20, 30, 40, 80 nodes; diamonds – nonuniform geometric grids with 10, 20, 30, 40, 80 nodes. Each solid line is a least squares linear fit of the form $a + bx$. Coefficient b defines the order of accuracy.

$$u(x) = \exp\left(-\frac{(x-2)^2}{0.1}\right). \tag{34}$$

The presence of the source term does not change the Jacobian matrix because $g \neq g(u)$. In addition to uniform grids, two types of nonuniform grids are also used here. One type of nonuniform grid (the so-called refined grid) is generated using the weight function of Eq. (30) with $\psi_1 = 4 \times 10^{-5}$ and $\psi_2 = 3.5 \times 10^{-7}$. This weight function produces a smooth clustering of nodes in the regions of strong gradients. Test calculations have been carried out with other forms of weight functions that give clustering of nodes in the regions of high spatial activity. No drastic differences in the comparative characteristics of the nonuniform grids have been noticed. Another type of nonuniform grid will be given below. A summary of the results of the compact scheme method on all three types of grids is given in Table 4.

The solutions of the compact scheme method for the uniform grids are plotted in Fig. 5. Only the solution on the smallest grid with 16 nodes departs significantly from the analytical profile. In Fig. 5, for purposes of comparison, a solution using the low order scheme is given on a uniform grid with 2000 nodes. That the quality of this low order solution is close to that of the compact scheme solution on the 16-node grid is remarkable, when one considers the two orders of magnitude difference in the number of nodes.

The compact scheme solutions for the nonuniform refined grids, plotted in Fig. 6, are also superior with respect to the solution of the low order scheme. The compact scheme solution on the 41-node grid produces almost the same results as the low order solution on the uniform grid with 2000 nodes. The compact scheme solution on the 81-node grid almost coincides with the analytical profile.

For a given number of nodes, one still notices that the solutions on the refined grids are less accurate than the solutions on the uniform grids. This fact is almost counterintuitive, as the refined grids are constructed so that more points are placed in the regions with high spatial activity. The idea of grid refinement has proven to be very effective in solutions with low order schemes [9,54]. However, there is one fundamental difference between the solutions with the compact scheme selected here and the solutions using the low order scheme. Low order schemes have the same overall order of accuracy for uniform and nonuniform grids, i.e., first order because of the convective term discretizations. In contrast, the compact scheme has overall third order accuracy on inner nodes for nonuniform grids, and the accuracy order rises to six on the inner nodes for uniform grids. The discretization order at the boundary nodes does not grow in the transition from nonuniform grids to uniform grids, but the bell-shaped profile is located away from the

Table 4
Bell-shaped profile problem

Run	Grid	Nodes	$h_{\max}$	$h_{\min}$	Absolute error
1	Uniform	16	0.333333	0.333333	0.0993206
2	Uniform	21	0.25	0.25	0.0206067
3	Uniform	41	0.125	0.125	0.000275771
4	Uniform	81	0.0625	0.0625	3.6541×10^{-6}
5	Uniform	161	0.03125	0.03125	5.5193×10^{-8}
6	Refined	16	0.500453	0.0863676	1.01252
7	Refined	21	0.37534	0.0646443	1.10796
8	Refined	41	0.187672	0.0305125	0.107468
9	Refined	81	0.0938373	0.014377	0.0050333
10	Refined	161	0.0469203	0.00657941	0.000343549
11	Composite	20	0.335088	0.25	0.0206063
12	Composite	37	0.248935	0.125	0.000275769
13	Composite	70	0.178864	0.0625	3.65413×10^{-6}
14	Composite	131	0.161774	0.03125	5.51317×10^{-8}

Summary of the results for calculations with the compact scheme on uniform and nonuniform grids for computational domain $L = 5$.

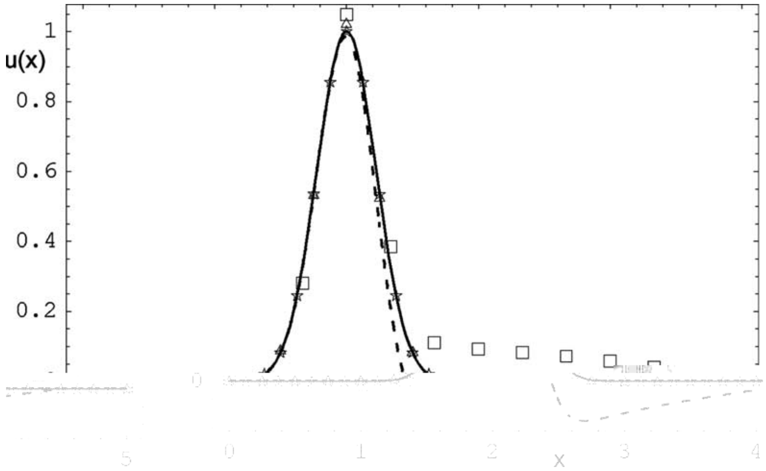

Fig. 5. Solution of the Burgers equation on uniform grids. Boxes – compact scheme on 16-node grid. Triangles – compact scheme on 21-node grid. Stars – compact scheme on 41-node grid. Analytical solution is shown with a solid line. Dashed line – low order scheme on the uniform grid with 2000 nodes.

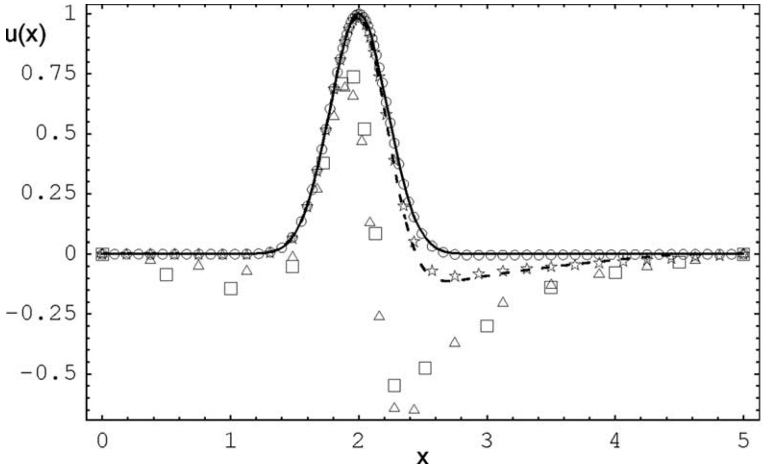

Fig. 6. Solution of the Burgers equation on nonuniform refined grids. Boxes – compact scheme on 16-node grid. Triangles – compact scheme on 21-node grid. Stars – compact scheme on 41-node grid. Circles – compact scheme on 81-node grid. Analytical solution is shown with a solid line. Dashed line – low order scheme on the uniform grid with 2000 nodes.

boundaries, so the role of boundary closures is minimal for the compact scheme accuracy for the given problem. Therefore, the higher order accuracy of the compact scheme on the inner nodes can be fully exploited, as follows. In regions with high spatial activity, a uniform node distribution can be used to tap the higher accuracy of the inner node discretizations of the compact scheme. For other parts of the computational domain, a nonuniform node distribution can be applied. This type of nonuniform grid will be called a composite grid.

Based on uniform grids (Table 4), a family of composite grids is constructed in the following manner. In the region $0 \leqslant x \leqslant 3.5$, uniform node distributions are used with the same step sizes as for the cases of the uniform grids given in Fig. 5. In the region $3.5 \leqslant x \leqslant 5$, geometric node distributions are used with a step size increase toward the right boundary: $h_{i+1} = 1.1h_i$. The results of the calculations on the composite grids have all the advantages of the uniform grid solutions but with fewer nodes. The maximum absolute error of each composite grid solution is very close to the error of the corresponding uniform grid solution (viz. Table 4).

The order of accuracy of the compact scheme is calculated for the bell-shaped profile on the three types of grids presented in Table 4. The decimal logarithm of the grid step size, $\log(h_{\min,m})$, vs. the decimal logarithm of the maximum absolute error, $\log(\varepsilon_m)$, is plotted in Fig. 7. The linear fits for the solution data are defined in the form $a + bx$ using least squares. The solution on the nonuniform refined grids shows fourth order accuracy ($b = 3.27$). This same fourth order accuracy has been defined for the compact scheme on the nonuniform random grids for the steady-state problem presented earlier (viz. Fig. 4). However, the compact scheme solution on the uniform grids shows a significant gain in accuracy as compared to the first steady-state case. Sixth order accuracy is obtained ($b = 6.13$) for the uniform grids. The same sixth order accuracy is observed for the solutions on the composite grids ($b = 6.18$).

To conclude our study of the composite grid strategy, we consider again the bell-shaped profile problem (Eq. (34)), but with the computational domain now extended to $L = 50$. This configuration resembles the domain configuration found in flame calculations, in which the domain size is much larger than the region of high gradients within the solution. For the extended domain, composite grids are again built of two zones: a uniform grid zone ($0 \leqslant x \leqslant 3.5$) and the geometric grid zone ($3.5 < x \leqslant 50$). In the geometric zone, the grid step grows according to $h_{i+1} = \gamma h_i$. For comparison purposes, calculations are also performed on uniform grids.

The parameters of the uniform and composite grids are presented in Table 5. The number of nodes in each composite grid $N^{\mathrm{c}}_{\mathrm{nodes}}$ is significantly less than in the uniform grid $N^{\mathrm{u}}_{\mathrm{nodes}}$ for the same minimum grid step size $h_{\min}$. Based on the maximum absolute error $\varepsilon_{\max}$ (viz. Table 5) and the minimum grid step size, the following fits are found for the data: for uniform grids, $\log_{10}\varepsilon_{\max} = 3.04 + 6.82\log_{10} h_{\min}$; and for composite grids, $\log_{10}\varepsilon_{\max} = 3.10 + 6.87\log_{10} h_{\min}$. Therefore, at least sixth order solution accuracy with the implicit compact scheme method is recovered for both uniform and nonuniform (composite) grids. We further

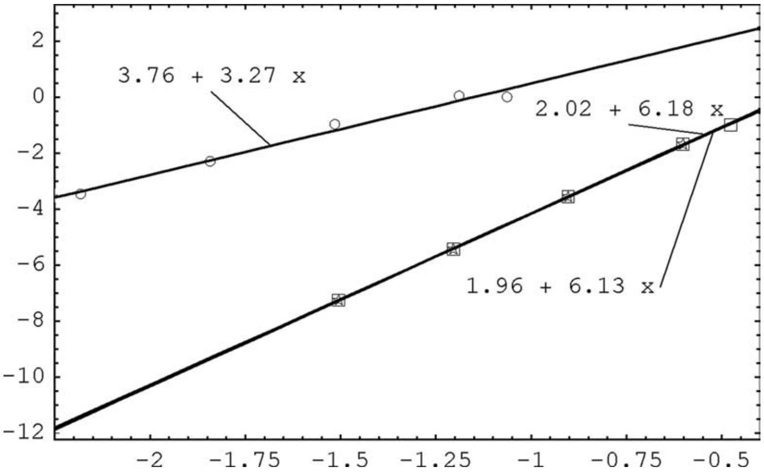

Fig. 7. Decimal logarithm of the maximum absolute error of the bell-shaped profile solution of the Burgers equation obtained using compact schemes vs. decimal logarithm of the maximum grid step size. Solution data is represented with symbols. Boxes – uniform grids with 16, 21, 41, 81, 161 nodes; circles – nonuniform refined grids with 16, 21, 41, 81, 161 nodes; stars – nonuniform composite grids with 20, 37, 70, 131 nodes. Each solid line is a least squares linear fit of the form $a + bx$. Coefficient b defines the order of accuracy.

note that no loss of accuracy is found for the given problem with the reduction of grid nodes in the region with insignificant gradients. The composite grid strategy results in an overall reduction factor in the number of nodes of between 5.91 (coarse grids) and 10.1 (fine grids), in comparison with uniform grids (viz. last column of Table 5).

3.2.1. Spectral properties of Jacobian matrices

Due to the relatively small dimension of the linear problem in the case of the one-dimensional Burgers equation, the spectral properties of the Jacobian matrix can be easily analyzed. These results may provide necessary information for the development of an implicit compact scheme solver for two-dimensional problems and may help develop efficient iterative solvers based on Krylov subspace methods [42].

The uniform grids of runs 4 and 5 of Table 4 are used with two function distributions: the analytical solution $u_a(x)$ (Eq. (34)) and a function obtained by multiplying the analytical solution function by 0.5, i.e., $0.5u_a(x)$. The latter case mimics a situation frequently encountered during pseudo-time stepping, when the function distribution deviates significantly from the solution. For each distribution, a compact scheme Jacobian matrix J is generated. Then, the norm of the residuals of the steady-state equation $\|f\|$ are computed as

$$\|f\| = \sqrt{\frac{f^2}{N}}. \tag{35}$$

Condition numbers $\kappa_1(J)$ and $\kappa_2(J)$ are defined via

$$\kappa_n(J) = \|J\|_n\|J^{-1}\|_n, \tag{36}$$

based on the n-norm of the Jacobian matrix. The same formula for the condition number is applied to the low order Jacobian J_{low}. Table 6 presents residual norms, condition numbers, as well as minimum and maximum eigenvalues for several discretized problems.

For the function distribution $0.5u_a(x)$, the steady-state residual norms are approximately the same for all cases with coarse (81-node) and fine (161-node) grids and for the low order and compact scheme solvers. For the function distribution $u_a(x)$, the residual norm is reduced consistently with the discretization order. The accuracy order of the spatial derivatives does not directly affect the size of the condition number. In the given solution of the Burgers equation, the condition numbers of J can be even less than the condition numbers of J_{low}, e.g., compare cases 6 and 8. Also, the size of the residual norm $\|f\|$ does not provide any indication as to the condition number of the Jacobian, e.g., compare cases 5 and 6.

In all cases in Table 6, the Jacobian eigenvalues are real-valued. The eigenvalues are almost always positive, with the exception of cases 8 and 20. Condition numbers for J and J_{low} behave similarly as the time step size increases; the condition number first decreases, reaches some minimum, and then grows again.

Table 5
Bell-shaped profile problem

Run	Uniform grid		Composite grid					$N^{\mathrm{u}}_{\mathrm{nodes}}/N^{\mathrm{c}}_{\mathrm{nodes}}$
	$N^{\mathrm{u}}_{\mathrm{nodes}}$	$\varepsilon_{\max}$	$N^{\mathrm{c}}_{\mathrm{nodes}}$	$\varepsilon_{\max}$	$h_{\min}$	$h_{\max}$	γ	
1	201	0.127	34	0.136	2.50×10^{-1}	8.00	1.200	5.91
2	401	4.26×10^{-4}	52	4.27×10^{-4}	1.25×10^{-1}	7.77	1.197	7.71
3	601	4.12×10^{-5}	68	4.12×10^{-5}	8.33×10^{-2}	7.80	1.199	8.83
4	801	7.51×10^{-6}	83	7.49×10^{-6}	6.25×10^{-2}	7.97	1.205	9.65
5	1001	1.98×10^{-6}	99	1.97×10^{-6}	5.00×10^{-2}	7.71	1.197	10.1

Summary of the results for calculations with the compact scheme on uniform and composite grids for large computational domain $L = 50$.

Table 6
Spectral properties of Jacobians constructed with compact schemes and low order (superscript L) discretizations in the case of the bell-shaped profile problem

Case	Nodes	Δt	$\|f\|$	$\kappa_1(J)$	$\kappa_2(J)$	min $\lambda(J)$	max $\lambda(J)$
1	161	10^{-6}	0.4645	1.0008×10^{6}	1.0007×10^{6}	1	1.0007×10^{6}
2	161	10^{-3}	0.4645	1855.7	1709.0	1	1702.3
3	161	1	0.4645	7547.4	1680.0	1	703.3
4	161	10^{3}	0.4645	67946	26532	3.259×10^{-2}	702.3
5	161	ss	0.4645	67946	26532	3.259×10^{-2}	702.3
6	161	ss^{a}	1.594×10^{-7}	68889	27863	3.386×10^{-2}	702.6
7	161^{L}	ss	0.4588	59367	25230	2.103×10^{-2}	435.9
8	161^{L}	ss^{a}	5.914×10^{-2}	137277	63579	-1.007×10^{-2}	464.1
9	81	10^{-6}	0.4630	1.0002×10^{6}	1.0002×10^{6}	1	1.0002×10^{6}
10	81	10^{-3}	0.4630	1213.1	1176.6	1	1175.7
11	81	1	0.4630	1020.2	316.0	1	176.7
12	81	10^{3}	0.4630	8761.4	6519.8	3.259×10^{-2}	175.7
13	81	ss	0.4630	8761.4	6519.8	3.259×10^{-2}	175.7
14	81	ss^{a}	1.061×10^{-5}	9104.8	6903.3	3.385×10^{-2}	176.0
15	81^{L}	10^{-6}	0.4549	1.0001×10^{6}	1.0001×10^{6}	1	1.0001×10^{6}
16	81^{L}	10^{-3}	0.4549	1146.9	1115.3	1	1114.5
17	81^{L}	1	0.4549	670.5	206.8	1	115.5
18	81^{L}	10^{3}	0.4549	18017	15470	8.796×10^{-3}	114.5
19	81^{L}	ss	0.4549	18017	15470	8.796×10^{-3}	114.5
20	81^{L}	ss^{a}	0.1136	5346.8	2952.2	-6.075×10^{-2}	127.8

Two uniform grids are considered: 81-node and 161-node. Jacobians are formed for pseudo-time and steady-state (labeled as ss) equations. Two distributions are considered: analytical $u_a(x)$ (labeled with superscript 'a') and $0.5u_a(x)$. Evaluation of the residual norm, condition numbers, and eigenvalues is explained in the text.

This behavior is in contrast with the monotone decrease of a condition number with decrease in Δt in the spectral analysis of low order Jacobians for a two-dimensional Poisson equation with periodic boundary conditions [16]. The following estimates for J (same applies to J_{low}) can be used for the problem considered here:

$$\kappa_n(J) \approx \Delta t^{-1}, \qquad \rho(J) = max|\lambda(J)| \approx \Delta t^{-1} \quad \text{for } \Delta t \leqslant 10^{-3}. \tag{37}$$

This eigenvalue behavior can be easily understood with J_{low}. Because of the Dirichlet boundary conditions, the characteristic equation for the Jacobian can be cast in the form

$$\det(J_{\mathrm{low}} - \lambda I) = \det\begin{pmatrix} 1-\lambda & 0 & 0 \\ 0 & T-\lambda I_T & 0 \\ 0 & 0 & 1-\lambda \end{pmatrix} = (1-\lambda)^2 \det(T-\lambda I_T) = 0, \tag{38}$$

where T is the tridiagonal submatrix (for three-point stencils), each of whose subdiagonal entries are a, main diagonal entries are b, and superdiagonal entries are c; I_{T} – the identity matrix of compatible size $N_{\mathrm{T}} = N - 2$; and λ – eigenvalues. The eigenvalues for the tridiagonal matrix T are [45]

$$\lambda_j = b + 2\sqrt{ac}\cos\left(\frac{\pi j}{N_T+1}\right), \quad j = 1,\ldots,N_T. \tag{39}$$

When $b \approx \Delta t^{-1} \gg \sqrt{ac}$, as in the case of small time steps, the eigenvalues are clustered around $\lambda_b \approx \Delta t^{-1}$, except two (small) values at $\lambda = 1$. If the time step is large, or effectively infinite as in the steady-state case, then, due to the presence of the second derivative operator, the constants a, b, and c are all $\mathrm{O}(h^{-2})$. The spectral radius of J_{low} is then

$$\rho(J_{\text{low}}) = O(h^{-2}). \tag{40}$$

For sufficiently large grids, $N_T \gg 1$, and the eigenvalues are bounded by

$$\min(1, b - 2\sqrt{ac}) \leqslant \lambda_j \leqslant b + 2\sqrt{ac}. \tag{41}$$

In the case of the compact scheme discretization, submatrix T is dense, which complicates the analysis. However, if the entries of matrix T are of the form $a_{ij}h^{-2}$, then an estimation similar to that of Eq. (40) can be expected for the spectral radius. As is evident from Table 6, in the case of dominating spatial operators, the spectral radius of J and of J_{low} for the Burgers equation grows four-fold when the grid step size is cut in half (compare cases 8, 20 and 5, 13 and 6, 14).

A large condition number frequently indicates the deterioration of the convergence of an iterative solver [55]. However, it is currently accepted that the clustering of eigenvalues is another important factor for the loss or gain of iterative solver convergence for general nonsymmetric matrices [56]. The distribution of the eigenvalues for J and J_{low} for several values of time step size are given in Fig. 8 for cases 1 through 8 of Table 6. As shown in Fig. 8, despite a large condition number in case 1, the eigenvalues are well clustered around minimum and maximum values, so, good convergence of the iterative solver for the linear system (e.g., Eq. (25)) is expected. With increased time step size, the clustering deteriorates; thus, an efficient preconditioner [42,56] may be a necessary feature of the iterative solver.

3.2.2. Implementation of partial Jacobians as preconditioners

A uniform 161-node grid (viz. Table 4) is used to investigate the effect of different sparsification criteria in order to form partial Jacobians. The width of each partial stencil (number of nodes) is plotted in Fig. 9 for each grid node and for the three sparsification strategies given by Eqs. (15)–(17). A constant width of the partial stencil for the inner nodes is dictated by the use of uniform grids. For nonuniform grids, the partial stencil width changes even among the inner nodes. In the bulk of the computational domain, the sparsification based on the Jacobian entries (Eq. (17)) produces a width close to the one given by the second

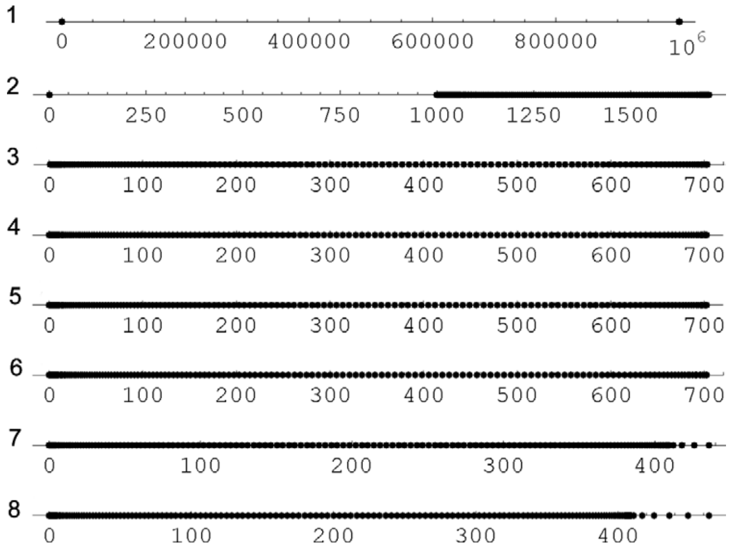

Fig. 8. Distribution of the Jacobian eigenvalues (real-valued) for several Jacobian matrices corresponding to cases 1 through 8 of Table 6.

derivative matrix (Eq. (16)). In the region where the convective term of the Burgers equation is not negligible, the width of the partial stencil is comparable to the one obtained with Eq. (15).

For the preconditioner equation

$$P\widehat{\Delta u} = -f \tag{42}$$

corresponding to Eq. (25), one may use $P = J_{\mathrm{p}}$. Currently, in the literature, there is no single a priori indicator of preconditioner quality. However, it is generally acknowledged that a good preconditioner results in a small condition number for the matrix product $P^{-1}J$. Ideally, $\kappa_n(J_{\mathrm{p}}^{-1}J) \approx 1$. In addition, the number of nonzeroes in J_{p} should be relatively small.

To study the condition number of product $J_{\mathrm{p}}^{-1}J$, the Jacobian matrix J of case 6 of Table 6 is selected. Shown in Fig. 10 is the difference $\Delta_\kappa = \kappa_2(J_{\mathrm{p}}^{-1}J) - 1$ vs. the nonzero density of J_{p} (number of matrix nonzeroes normalized by the total number of matrix entries). We select two sparsification methods. Method 1 is given by Eq. (17). Method 2 is based on Eqs. (15) and (16); a ‘‘greedy’’algorithm is used to select the wider stencil of the two possible. For each sparsification algorithm, seven threshold values are used: $\theta = 10^{-k}$, for $k = 1, \ldots, 7$. Fig. 10 also contains a single data point for the case when J_{low} (case 8 of Table 6) is used as the preconditioner. Preconditioning by J_{low} is advocated in [57] for a second-order-accurate implicit solver. However, when compact schemes are used with at least third order accuracy, then a low order discretization may not produce an efficient preconditioner. In the case considered, the use of J_{low} as a preconditioner resulted in a 60-fold reduction in the condition number, as compared to the condition number $\kappa_2(J)$ given

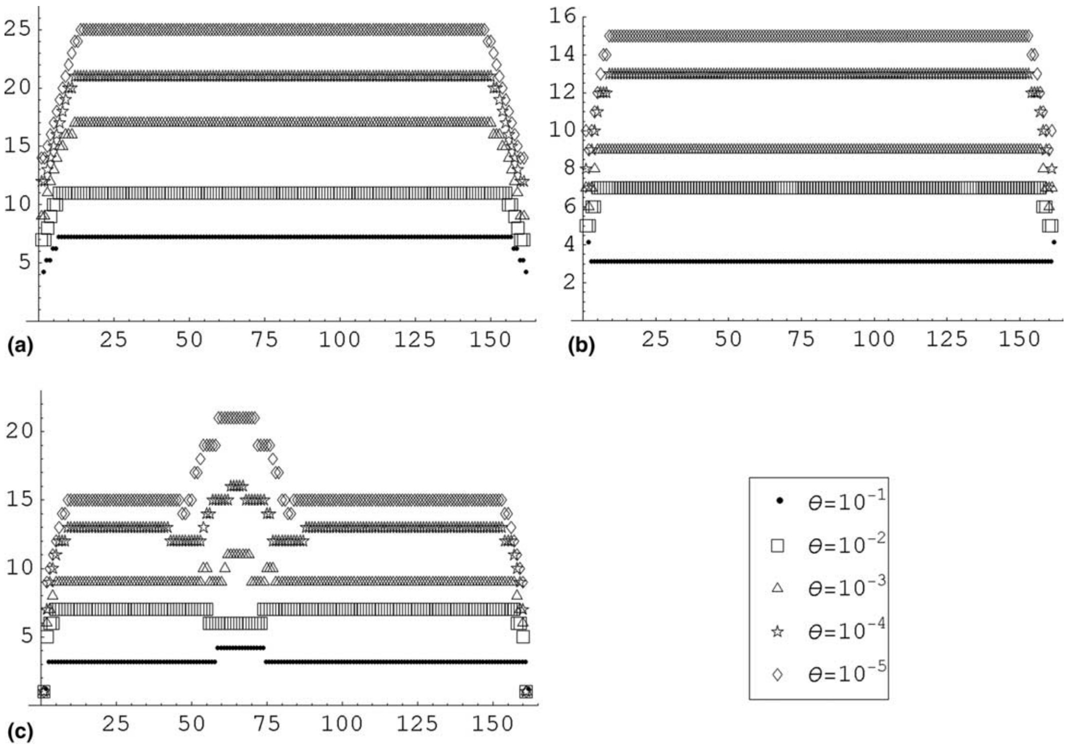

Fig. 9. Width of partial stencils throughout the computational domain. Number of grid nodes: 161. Partial stencils are selected based on several criteria for threshold θ. Panel (a) corresponds to Eq. (15), panel (b) – Eq. (16), panel (c) – Eq. (17).

in Table 6. However, the condition number $\kappa_2(J_{\mathrm{low}}^{-1}J)$ may still be large enough to desire a better preconditioner.

Comparison of Methods 1 and 2 for the sparsification of the Jacobian shows that for a given nonzero density of the preconditioner, Method 1 results in almost an order of magnitude reduction in the difference Δ_κ in comparison with Method 2. From an implementation point of view, Method 1 may be less efficient, as it requires the dynamic analysis of Jacobian entries each time the preconditioner is formed, and it may require the generation of more Jacobian entries than would normally be stored. In Method 2, which results in a static sparsity pattern for the partial Jacobian, the analysis of the stencil matrices must be done only once for each grid.

Despite the differences in implementation, both methods systematically reduce of the condition number based on threshold value. The difference Δ_κ can be made as small as needed with the use of smaller thresholds. Of course, smaller thresholds result in increased J_{p} densities. The optimal threshold values in Eqs. (15)–(17) may vary based on the requirements for solution accuracy, the size of the grid, the number of unknowns per grid node, and the parameters of the computer equipment. Another factor of uncertainty for the optimal threshold is the choice for the inexact (incomplete) decomposition of the partial Jacobian. The difference Δ_κ in Fig. 10 represents the theoretical maximum of the preconditioning strategy that can be achieved for a generated partial Jacobian.

3.3. Solution of the time-dependent Burgers equation

The linear ($w = c$) Burgers equation (Eq. (24)) has the time-dependent solution

$$u(x,t) = \exp(-\varphi^2\mu t)\sin(\varphi(x-ct)), \tag{43}$$

when provided with an initial condition of the form $u(x,0) = \sin(\varphi x)$ and compatible boundary conditions

$$u(x_{1,N},t) = \exp(-\varphi^2\mu t)\sin(\varphi(x_{1,N}-ct)) \tag{44}$$

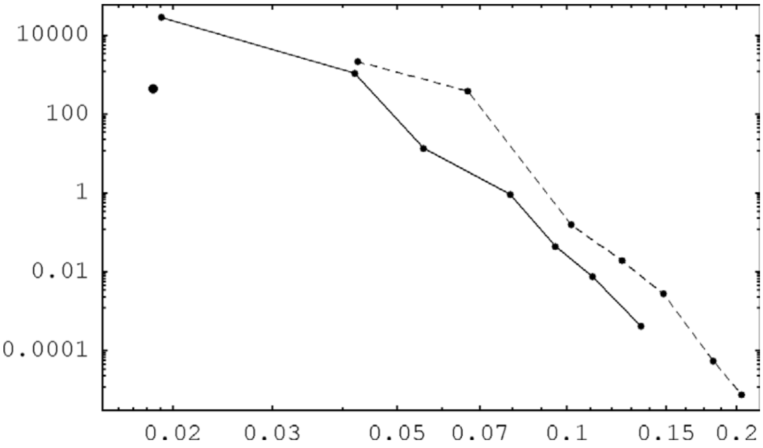

Fig. 10. Effect of the sparsification method on the quality of the preconditioner. Horizontal coordinate: sparsity of a partial Jacobian. Vertical coordinate: $\kappa(J_{\mathrm{p}}^{-1}J) - 1$. Solid line – sparsification based on the absolute value of Jacobian entries, dashed line – sparsification based on the partial stencils (“greedy” algorithm), and large solid dot – preconditioning by a low order Jacobian. For each sparsification method, seven threshold values are used: $\theta = 10^{-k}$, $k = 1, \ldots, 7$. Larger threshold values correspond to smaller sparsity of partial Jacobian.

on the domain $x_1 \leqslant x \leqslant x_N$. The solution models the exponential decay of a propagating harmonic wave, for which the direction of propagation is from left to right ($c,\varphi > 0$). The problem parameters are: $c = \varphi = 1$; $\mu = 0.5$; $x_1 = 0$; and $x_N = 6\pi$. Only uniform grids and constant time steps are used for the calculations. The solutions are obtained on the time interval $0 < t \leqslant 10$. The compact scheme method and the low order method have been employed, together with one of three implicit temporal discretization methods: the first-order Euler (EU), the second order Crank–Nicolson (CN), and second order backward differences (B2). A summary of the time-dependent calculations is given in Table 7. Calculations are performed on a computer system equipped with an AMD Opteron 240 (1.4 GHz) processor. The total calculation time is given as t_{CPU} in Table 7. On small grids, the calculation time was limited mainly by the rate of data file output.

The error norm $\|\Delta u^k\|$ of the time-dependent calculations is evaluated according to Eq. (35), where the absolute error $\Delta u^k = u_{\text{num}}(x_i,t^k) - u(x_i,t^k)$ is known for each time level t^k. Here, $u_{\text{num}}(x_i,t^k)$ represents the numerical solution vector defined on the grid at time level t^k; $u(x_i,t^k)$ – analytical solution vector defined on the same grid for the same time level; and N – number of nodes in the grid. The maximum error norm for each time-dependent solution is given in the last column of Table 7. Results of runs 3, 4, 5, 10, 14, and 17 are plotted in Figs. 11–13 for solution times $t = 1$, 5, and 10. Linear interpolation is used between the two closest computed time levels to obtain a solution for the selected time.

Based on the solutions presented for times $t = 1$, 5, and 10, the following observations can be made. At early times, the solutions with all methods closely follow the analytical profile. Only small deviations of the solution for the low order scheme on a coarse grid (viz. Fig. 11) become noticeable. At intermediate solution times, the deviation of the low order scheme solution becomes more apparent (viz. Fig. 12), and in addition, the deviations of the low order scheme solution on the fine grid (run 17) and of the compact scheme method with the Euler time discretization (run 4) appear. In the final plot (Fig. 13), only the solution of the compact scheme with the Crank–Nicolson time discretization (run 5) still follows the analytical solution.

Table 7
Time-dependent Burgers equation

Run	Spatial scheme	Nodes	Time scheme	Δt	t_{CPU}, s	$\max_k\|\Delta u^k\|$	$\max_k\|\Delta\bar{u}^k\|$
1	Compact	21	EU	9.42477×10^{-2}	0.05	2.682×10^{-2}	0.2993
2	Compact	21	CN	9.42477×10^{-2}	0.03	5.407×10^{-3}	0.2123
3	Compact	21	B2	9.42477×10^{-2}	0.06	5.703×10^{-3}	0.2212
4	Compact	41	EU	4.44132×10^{-2}	0.11	1.311×10^{-2}	0.1970
5	Compact	41	CN	4.44132×10^{-2}	0.11	4.133×10^{-4}	0.02587
6	Compact	41	B2	4.44132×10^{-2}	0.10	5.889×10^{-4}	0.02165
7	Compact	81	EU	1.11033×10^{-2}	1.06	3.378×10^{-3}	0.05294
8	Compact	81	CN	1.11033×10^{-2}	1.04	2.739×10^{-5}	0.001791
9	Compact	81	B2	1.11033×10^{-2}	1.13	3.749×10^{-5}	0.001473
10	Low order	41	EU	4.44132×10^{-2}	0.08	9.514×10^{-2}	1.604
11	Low order	41	CN	4.44132×10^{-2}	0.08	8.980×10^{-2}	1.605
12	Low order	41	B2	4.44132×10^{-2}	0.15	8.843×10^{-2}	1.567
13	Low order	81	EU	1.11033×10^{-2}	0.64	5.195×10^{-2}	0.9693
14	Low order	81	CN	1.11033×10^{-2}	0.80	5.036×10^{-2}	0.9631
15	Low order	81	B2	1.11033×10^{-2}	0.58	5.019×10^{-2}	0.9585
16	Low order	401	EU	4.44132×10^{-4}	138.08	1.124×10^{-2}	0.2384
17	Low order	401	CN	4.44132×10^{-4}	136.46	1.116×10^{-2}	0.2376
18	Low order	401	B2	4.44132×10^{-4}	138.77	1.116×10^{-2}	0.2376

Summary of the results for calculations with the compact scheme and low order methods on uniform grids.

In Fig. 14, the decimal logarithm of the error norm $\|\Delta u^k\|$ is plotted against the computed time. The error of the low order scheme exhibits linear decay with grid spacing reduction. There is almost no difference in the results obtained with first or second order temporal discretization as error lines almost coincide for runs 10, 11, and 12, as well as for runs 13, 14, and 15 and for runs 16, 17, and 18. The error of the compact scheme method in combination with the first order Euler scheme decays linearly with the reduction of grid spacing (runs 1, 4, and 7 (viz. Table 7)). Thus, the Euler discretization produces the largest error. If a second order temporal discretization is used, together with the compact schemes, then approximately a fourth order spatial accuracy is achieved. The coefficients of the linear fit $\{\log(h_{\max,m}), \log(\max\|\Delta u^k\|)\}$ are 3.81 and 3.62 for Crank–Nicolson and second-order backward difference schemes, respectively (viz. Table 7). Between the two second order temporal discretizations, the Crank–Nicolson scheme results in somewhat smaller computational error for the problem selected. Additionally, in Table 7, the maximum of a relative solution error $\|\Delta\bar{u}^k\|$ is given, in which Δu^k is scaled with $\exp(-\varphi^2\mu t)$ at each time level.

As shown in Table 7, the CPU time of the low order accuracy code must be several orders of magnitude larger than the CPU time of compact scheme code in order to obtain the same temporal and spatial accuracy. The following reasoning justifies this conclusion. The low order accuracy code has an error of the form $\mathrm{O}(\Delta x_{\mathrm{low}}) \sim \mathrm{O}(N_{\mathrm{low}}^{-1})$. The error for the compact scheme code is $(\Delta x_{\mathrm{cs}}^k) \sim \mathrm{O}(N_{\mathrm{cs}}^{-k})$. Here, Δx_{low} and Δx_{cs} are the grid spacings for low order and compact scheme codes, N_{low} and N_{cs} are the number of grid nodes in the same codes, and $k \geqslant 3$ is the overall order of spatial accuracy of the compact scheme in use. In order to obtain the same accuracy of the solutions of low order and compact scheme codes, the following must be satisfied

$$a_{\mathrm{cs}}N_{\mathrm{cs}}^{-k} = a_{\mathrm{low}}N_{\mathrm{low}}^{-1}, \tag{45}$$

where a_{cs} and a_{low} are some coefficients which depend on the form of the solution and the coefficients inside the expressions for local truncation errors of the corresponding discretizations. For sufficiently refined grids, the time step is limited by the diffusion operator discretization. If a total flow time t must be simulated, then the low order code must take $M_{\mathrm{low}} = \mathrm{O}(t\Delta x_{\mathrm{low}}^{-2}) \sim \mathrm{O}(tN_{\mathrm{low}}^2)$ steps and the compact scheme code

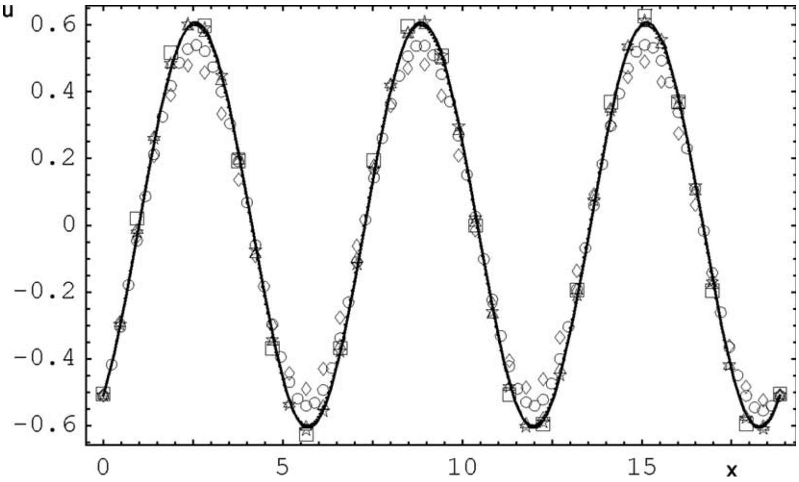

Fig. 11. Solution of the time-dependent Burgers equation at time t = 1 for the runs presented in

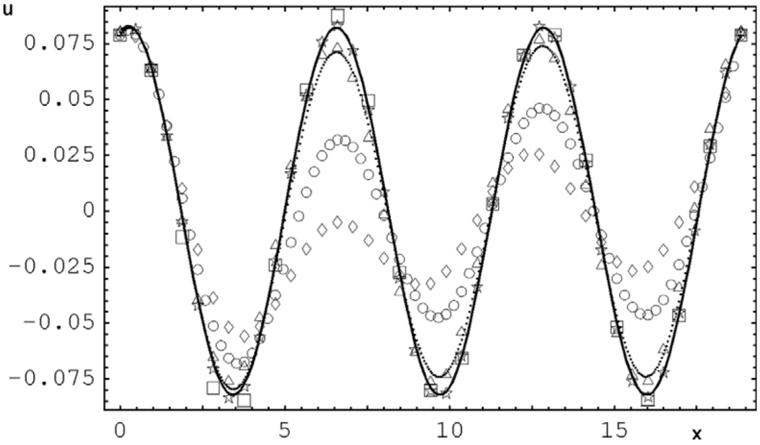

Fig. 12. Solution of the time-dependent Burgers equation at time $t = 5$ for the runs presented in Table 7. The legend is the same as in Fig. 11.

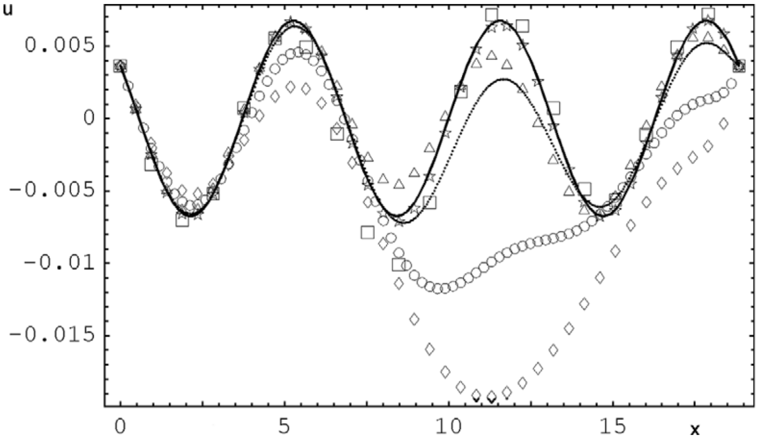

Fig. 13. Solution of the time-dependent Burgers equation at time $t = 10$ for the runs presented in Table 7. The legend is the same as in Fig. 11.

must take $M_{\mathrm{cs}} = \mathrm{O}(t\Delta x_{\mathrm{cs}}^{-2}) \sim \mathrm{O}(tN_{\mathrm{cs}}^2)$ steps. At each time level, the most expensive operation within the implicit compact scheme solver is the solution of the linear system, which requires the Gaussian decomposition of the dense Jacobian matrix. The cost of this operation expressed in floating point operations is $\mathrm{cost}_{\mathrm{cs}} = \mathrm{O}\left(\frac{N_{\mathrm{cs}}^3}{3}\right)$. The low order Jacobian has a tridiagonal structure and the solution of the corresponding linear system can be effectively accomplished with the Thomas algorithm, of $\mathrm{cost}_{\mathrm{cs}} = \mathrm{O}(3N_{\mathrm{low}})$. To leading order, the floating point operation count for the compact scheme solver to model the solution for a total time t is

$$\mathrm{count}_{\mathrm{cs}} = \mathrm{cost}_{\mathrm{cs}} M_{\mathrm{cs}} = \mathrm{O}\left(\frac{1}{3}N_{\mathrm{cs}}^3 M_{\mathrm{cs}}\right). \tag{46}$$

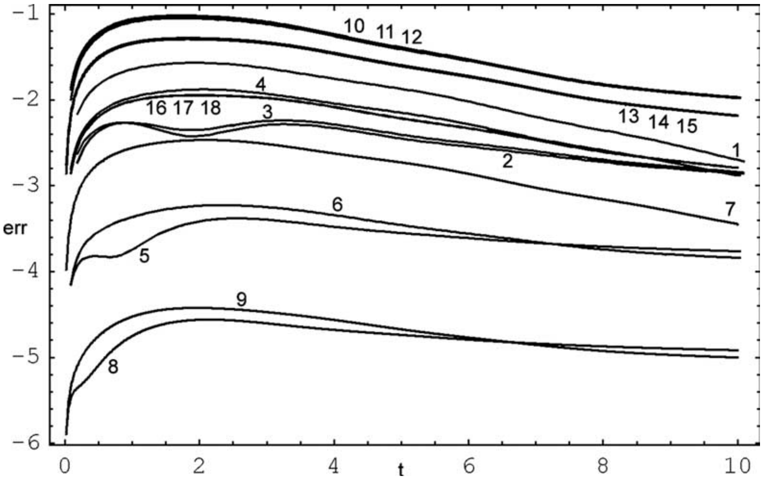

Fig. 14. Decimal logarithm of maximum error norm vs. the computed time for solutions with the compact scheme and the low order methods. Descriptions of runs 1 through 18 are given in Table 7.

According to Eq. (45), in order to obtain the same solution accuracy, the number of grid nodes of the low order solver must be $N_{\text{low}} = \frac{a_{\text{low}} N_{\text{cs}}^k}{a_{\text{cs}}}$. For a given total time t, a given domain length, and given coefficients of the Burgers equation (thus leading to a given Fourier number value), we can write $M_{\text{low}} = M_{\text{cs}} N_{\text{low}}^2 N_{\text{cs}}^{-2}$. Only taking into account the linear solve within the implicit solver, the operation count for the low order solver to model the solution for a total time t is

$$\text{count}_{\text{low}} = \text{cost}_{\text{low}} M_{\text{low}} = \text{O}\left(3N_{\text{low}} M_{\text{cs}} \frac{N_{\text{low}}^2}{N_{\text{cs}}^2}\right) = \text{O}\left(3\left(\frac{a_{\text{low}}}{a_{\text{cs}}}\right)^3 N_{\text{cs}}^{3k-2} M_{\text{cs}}\right). \tag{47}$$

If, for a given problem, the compact scheme achieves fourth order accuracy in space ($k = 4$), then we have

$$\text{count}_{\text{low}} = \text{O}\left(3\left(\frac{a_{\text{low}}}{a_{\text{cs}}}\right)^3 N_{\text{cs}}^{10} M_{\text{cs}}\right). \tag{48}$$

The comparison of operation count for the compact scheme solver (Eq. (46)) and the low order solver (Eq. (48)) shows that even with the different algorithms used for the decomposition of Jacobian matrices, the compact scheme solver is always more efficient for time-dependent problems than the low order solver. This fact can explain the several orders of magnitude difference observed in CPU time for runs 2 and 17, with still much lower accuracy of the low order solver.

3.4. Unstable flame propagation

Finally, we demonstrate the ability of the compact scheme solver to model stiff problems. A well-known unstable flame [58] (test problem A) is considered. Similar problems of unstable flame propagation were considered in [59,60]. A propagating planar flame is modeled with a single step reaction. Under the conditions of nonunity Lewis number ($Le > 2$), oscillations in flame velocity may occur. A detailed description of this problem can be found in [58]. Here for brevity, we provide only a mathematical formulation of the problem. A system of two PDEs must be solved simultaneously:

$$\frac{\partial T}{\partial t} - \frac{\partial^2 T}{\partial x^2} = R, \tag{49}$$

$$\frac{\partial Y}{\partial t} - \frac{1}{Le}\frac{\partial^2 Y}{\partial x^2} = -R, \tag{50}$$

where T is the normalized temperature and Y is the normalized mass fraction of the reactant. The normalized reaction rate R is defined as

$$R = \frac{\beta^2 Y}{2Le}\exp\left(\frac{\beta(T-1)}{1+\alpha(T-1)}\right). \tag{51}$$

Reaction rate coefficients and the Lewis number are given by $\alpha = 0.8$, $\beta = 20$, and $Le = 2$. The initial conditions are $T = \exp(x), Y = 1 - \exp(xLe)$ for $x \leqslant 0$ and $T = 1$, $Y = 0$ for $x > 0$. The boundary conditions are $T = 0$, $Y = 1$ for $x \to -\infty$ and $\partial T/\partial x = \partial Y/\partial x = 0$ for $x \to \infty$. This problem setup defines Case 4 of [58, p. 10]. An accurate solution of this problem requires a very stable numerical algorithm to calculate a set of stiff PDEs without the introduction of numerical instabilities. In addition, a numerical scheme must have excellent spatial resolution properties in order to model the flame velocity evolution.

For our calculations we use the implicit compact scheme solver with the Crank–Nicolson temporal discretization with a time step $\Delta t = 10^{-3}$. A uniform grid is used in the domain of interest where a flame front is located. To the left and right of the domain of interest we apply a geometric grid with a step size increase in the direction toward boundaries. The width of the domain of interest is 20. A set of composite grids is used which corresponds to the uniform grid spacing of $\Delta x_{\mathrm{di}} = 0.2$, 0.1, 0.05, and 0.025, respectively. In these grids, there are 101, 201, 401, and 801 grid nodes in the domain of interest. There are additional 20 to 50 grid nodes located outside of the domain of interest.

An evolution of the flame front velocity is shown in Fig. 15. In the case of underresolved grid, wiggles appear in the flame velocity evolution (curve *a*). We stress that no wiggles were observed in the instanteneous profiles of the temperature and species. As the domain of interest becomes more refined, we observe a flame evolution which is in full accordance with the results predicted in [58]. We estimate that for the low order solver employing centered differences one has to use grid spacing Δx at least one order smaller than that of Δx_{di} to observe a similar behavior of the unsteady flame (note the absence of convective terms in the governing equations).

4. Conclusions

In the present work, we have developed a novel, higher order in space, one-dimensional, implicit compact scheme solver on nonuniform grids, suitable for modeling steady-state and time-dependent problems. Ultimately, this solver is targeted as an alternative to traditional implicit solvers with low order spatial discretizations for the solution of stiff nonlinear time-dependent problems in multidimensional configurations. In order to obtain an efficient implicit solver, we have reviewed and analyzed several possible options for spatial discretizations and have shown several advantageous properties of compact schemes. We have applied necessary stability conditions to study the properties of the implicit compact scheme solver. In all cases considered, the implicit solver has been shown to be unconditionally stable.

Facing the lack of published algorithms suitable for the development of an efficient, implicit solver with compact scheme discretizations, we have defined three tasks that must be solved in order to obtain an accurate and relatively inexpensive treatment of the Jacobian matrices: how to generate a Jacobian matrix; how to evaluate a Jacobian-vector product; and how to solve a preconditioner equation. Based on the linearity of compact schemes, we have demonstrated how to determine entries in the Jacobian matrix and carry out

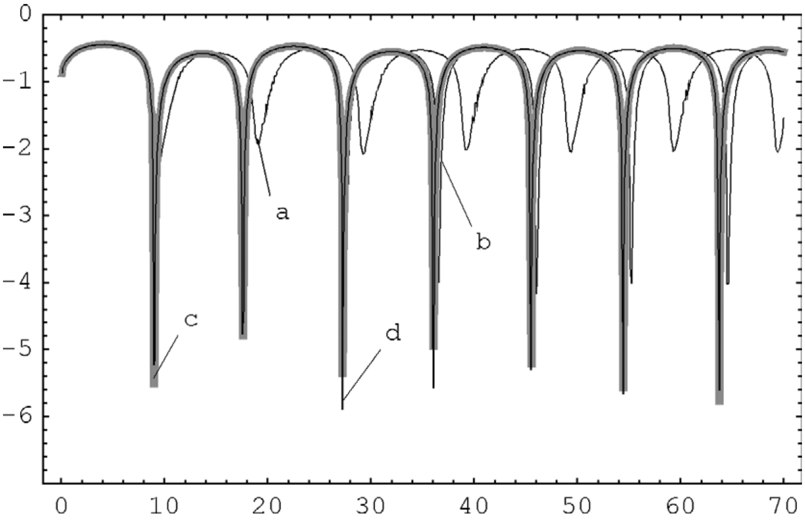

Fig. 15. Dimensionless unstable flame velocity vs. dimensionless flow time. Curves *a*, *b*, *c*, and *d* represent compact scheme solutions with $\Delta x_{\mathrm{di}} = 0.2, 0.1, 0.05$, and 0.025, respectively.

Jacobian-vector multiplication exactly, using operations involving the component form of the Jacobian. Based on the exponential decay of off-diagonal stencil matrix entries, we have indicated approaches for the systematic approximation (sparsification) of dense Jacobians constructed with compact schemes.

To demonstrate the advantages of the implicit compact scheme solver, we have solved several steady-state and time-dependent problems for the Burgers equation. In all problems considered, the implicit compact scheme solver has shown more than an order of magnitude (and, for some problems, more than two orders of magnitude) gain of accuracy in comparison with a traditional implicit low order solver.

An analysis of the implicit compact scheme solver has been carried out to demonstrate an order of spatial accuracy varying between three (on general nonuniform grids) and six (on uniform and composite grids). The spatial accuracy has been found to be affected by grid structure and by the presence of steep gradients near the boundary nodes. We have illustrated the benefits of using a composite form of nonuniform gridding that enables the implicit compact scheme solver to achieve sixth order accuracy. In our model time-dependent problem, several orders of magnitude reduction in computational time was achieved when the compact scheme solver was used instead of the low order solver. The new solver was found to be robust when solving a stiff system of PDEs which models an unstable flame propagation. For a well-known test flame problem, excellent agreement with previously published results has been recovered in a compact scheme solver solution.

Finally, an analysis of the spectral properties of the Jacobian matrices has been presented to show the behavior of condition numbers and eigenvalue distributions. The condition number of the compact scheme Jacobian has been found to be of comparable magnitude to that of the low order Jacobian for the same problem setup. This fact is viewed as encouraging and shows that there is currently no evidence found against the use of compact scheme Jacobians within iterative solvers in the case of multidimensional problems.

Acknowledgement

This work was supported in part by the DOE Office of Basic Energy Sciences and the National Science Foundation. The first author would like to thank Prof. M. Benzi of Emory University for discussions on linear algebra problems and Dr. B.A.V. Bennett of Yale University for stimulating discussions and help in proofreading this manuscript.

References

[1] C. Curtiss, J. Hirschfelder, Integration of stiff equations, Proc. Nat. Acad. Sci. 38 (1952) 235–243.

[2] J. Faires, R. Burden, Numerical Methods, third ed., Brooks/Cole Pub. Co, Pacific Grove, 2003.

[3] C. Lund, HCT: A general computer program for calculating time dependent phenomena involving one-dimensional hydrodynamics, transport, and detailed chemical kinetics, Report UCRL52504, Livermore National Laboratory, 1978.

[4] C. Westbrook, A generalized ICE method for chemically reactive flows in combustion systems, J. Comput. Phys. 29 (1978) 67–80.

[5] J. Warnatz, Numerical problems arising from the simulation of combustion phenomena, in: P. Deuflhard, B. Engquist (Eds.), Large Scale Scientific Computing, Birkhauser Boston, Cambridge, MA, 1985, pp. 51–74.

[6] G. Goyal, U. Maas, J. Warnatz, Simulation of the behavior of rich hydrogen–air flames near the flammability limit, Combust. Sci. Technol. 105 (1995) 183–193.

[7] M. Smooke, Solution of burner-stabilized premixed laminar flames by boundary-value methods, J. Comput. Phys. 48 (1982) 72–105.

[8] M. Smooke, J. Miller, R. Kee, On the use of Adaptive Grids in Numerically Calculating Adiabatic Flame Speeds in Numerical Methods in Laminar Flame Propagation, Friedr. Vieweg und Sohn, Braunschweig, 1982.

[9] M. Smooke, On the use of adaptive grids in premixed combustion, AIChE J. 32 (1986) 1233–1241.

[10] M. Smooke, I. Puri, K. Seshadri, A comparison between numerical calculations and experimental measurements of the structure of a counterflow diffusion flame burning diluted methane in diluted air, in: Twenty-First Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, 1986, pp. 1783–1792.

[11] M. Smooke, K. Seshadri, I. Puri, The structure and extinction of partially premixed flames burning methane in air, in: Twenty-Second Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, 1988, pp. 1555–1563.

[12] M. Smooke, J. Crump, K. Seshadri, V. Giovangigli, Comparison between experimental measurements and numerical calculations of the structure of counterflow, diluted, methane–air, premixed flames, in: Twenty-Third Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, 1990, pp. 463–470.

[13] M. Smooke, R. Mitchell, D. Keyes, Numerical solution of two-dimensional axisymmetric laminar diffusion flames, Combust. Sci. Technol. 67 (1989) 85–122.

[14] M. Smooke, V. Giovangigli, Numerical modeling of axisymmetric laminar diffusion flames, IMPACT Comput. Sci. Engng. 4 (1992) 46–79.

[15] Y. Xu, M. Smooke, Application of a primitive variable Newton's method for the calculation of an axisymmetric laminar diffusion flame, J. Comput. Phys. 104 (1993) 99–109.

[16] A. Ern, V. Giovangigli, D. Keyes, M. Smooke, Towards polyalgorithmic linear system solvers for nonlinear elliptic problems, SIAM J. Sci. Comput. 15 (1994) 681–703.

[17] B. Bennett, J. Fielding, R. Mauro, M. Long, M. Smooke, A comparison of the structures of lean and rich axisymmetric laminar Bunsen flames: Application of local rectangular refinement solution-adaptive gridding, Combust. Theory Modelling 3 (1999) 657–687.

[18] E. Burman, A. Ern, V. Giovangigli, An adaptive finite element method with crosswind diffusion for low Mach, steady, laminar combustion, J. Comput. Phys. 188 (2003) 472–492.

[19] B. Bennett, M. Smooke, Unsteady axisymmetric laminar diffusion flames: An application of local rectangular refinement, in: Proc. 2001 Technical Meeting of the Eastern States Section of the Combustion Institute, The Combustion Institute, Hilton Head, SC, 2001, pp. 332–335.

[20] B. Bennett, M. Smooke, Local rectangular refinement solution-adaptive gridding with application to unsteady combustion, in: Ninth International Conference on Numerical Combustion, SIAM and The Combustion Institute, Sorrento, Italy, 2002, pp. 213–214.

[21] B. Bennett, M. Smooke, Local rectangular refinement solution-adaptive gridding with application to oscillating laminar diffusion flames, in: Proc. 3rd Joint Meeting of the US Sections of the Combustion Institute, The Combustion Institute, Chicago, IL, 2003, p. B37.

[22] B. Leonard, A stable and accurate convective modeling procedure based on quadratic upstream interpolation, Comput. Meth. Appl. Mech. Engr. 19 (1979) 59–98.

[23] A. Harten, On a class of high resolution total-variation-stable finite-difference schemes, SIAM J. Numer. Anal. 21 (1) (1984) 1–23.

[24] J. Tannehill, D. Anderson, R. Pletcher, Computational Fluid Mechanics and Heat Transfer, second ed., Taylor and Francis, Washington, 1997.

[25] J. Zhang, An explicit fourth-order compact finite difference scheme for three-dimensional convection-diffusion equation, Commun. Numerical Meth. Eng. 14 (1998) 209–219.

[26] J. Zhang, Multigrid method and fourth-order compact scheme for 2D Poisson equation with unequal mesh-size discretization, J. Comput. Phys. 179 (2002) 170–179.

[27] W. Spotz, G. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int. J. Numer. Meth. Eng. 38 (1995) 3497–3512.

[28] W. Spotz, G. Carey, Formulation and experiments with high-order compact schemes for nonuniform grids, Int. J. Numer. Meth. Heat Fluid Flow 8 (1998) 288.

[29] S. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

[30] R. Wilson, A. Demuren, M. Carpenter, Higher-order compact schemes for numerical simulation of incompressible flows part II: applications, Numer. Heat Transfer Part B 39 (2001) 231–255.

[31] M. Noskov, An implicit compact scheme solver for modeling steady-state and time-dependent axisymmetric flows with application to laminar diffusion flame simulations, Ph.D. thesis, Yale University, New Haven, Mechanical Engineering Department, 2003.

[32] A. Tolstykh, Method of numerical solution of Navier–Stokes equations for a compressible gas in wide-range of Reynolds numbers, Doklady Akademii Nauk SSSR 210 (1973) 48–51, Engl. Transl., Soviet Physics. Doklady. 18:74, 1973.

[33] S. Orszag, M. Israeli, Numerical Simulation of Viscous Incompressible Flows, vol. 6, Annual Reviews, Inc, Palo Alto, CA, 1974, pp. 281–318.

[34] A. Tolstykh, M. Lipavskii, On performance of methods with third- and fifth-order compact upwind differencing, J. Comput. Phys. 140 (1998) 205–232.

[35] K. Mahesh, A family of high order finite difference schemes with good spectral resolution, J. Comput. Phys. 145 (1998) 332–358.

[36] L. Gamet, F. Ducros, F. Nicoud, T. Poinsot, Compact finite difference schemes on non-uniform meshes. Application to direct numerical simulations of compressible flows, Int. J. Numer. Meth. Fluids 29 (1999) 159–191.

[37] C. Jimenez, B. Cuenot, T. Poinsot, D. Haworth, Numerical simulation and modeling for lean stratified propane-air flames, Combust. Flame 128 (2002) 1–21.

[38] M. Visbal, D. Gaitonde, High-order accurate methods for complex unsteady subsonic flows, AIAA J. 37 (10) (1999) 1231–1239.

[39] M. Visbal, D. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, J. Comput. Phys. 181 (2002) 155–185.

[40] J. Pereira, M. Kobayashi, J. Pereira, A fourth-order-accurate finite volume compact method for the incompressible Navier–Stokes solutions, J. Comput. Phys. 167 (2001) 217–243.

[41] Y. Saad, A flexible inner–outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469.

[42] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., SIAM, Philadelphia, 2003.

[43] S. Demko, W. Moss, P. Smith, Decay rates for inverses of band matrices, Math. Comput. 43 (1984) 491–499.

[44] M. Carpenter, D. Gottlieb, S. Abarbanel, The stability of numerical boundary treatments for compact high-order finite-difference schemes, J. Comput. Phys. 108 (1993) 272–295.

[45] G. Smith, Numerical Solution of Partial Differential Equations, third ed., Clarendon Press, Oxford, 1985.

[46] E. Benton, G. Platzman, A table of solutions of the one-dimensional Burgers equation, Quart. Appl. Math. 30 (1972) 195.

[47] W. Briley, H. McDonald, Solution of the multi-dimensional compressible Navier–Stokes equations by a generalized implicit method, J. Comput. Phys. 24 (1977) 372–397.

[48] W. Briley, H. McDonald, An overview and generalization of implicit Navier–Stokes algorithms and approximate factorization, Comput. Fluids 30 (2001) 807–828.

[49] J. Dennis, R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall Inc, Englewood Cliffs, NJ, 1983.

[50] P. Deulfhard, Modified Newton method for solution of ill-conditioned systems of nonlinear equations with application to multiple shooting, Numer. Math. 22 (1974) 289–315.

[51] M. Smooke, Error estimate for the modified Newton method with applications to the solution of non-linear, 2-point boundary-value problems, J. Optim. Theory Appl. 39 (1983) 489–511.

[52] A. Hindmarsh, ODEPACK, a systematized collection of ODE solvers, IMACS Trans. Sci. Comput. 1 (1983) 55–64.

[53] R. LeVeque, Numerical Methods for Conservation Laws, second ed., Birkhauser Verlag, Basel, 1992.

[54] M. Smooke, R. Mattheij, On the solution of nonlinear two-point boundary value problems on successively refined grids, Appl. Numer. Math. 1 (1985) 463–487.

[55] R. Barrett, H.V. der Vorst, R. Pozo, J. Dongarra, V. Eijkhout, C. Romine, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1993.

[56] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2002) 418–477.

[57] W. Gropp, D. Keyes, L. McInnes, M. Tidriri, Globalized Newton–Krylov–Schwarz algorithms and software for parallel implicit CFD, Int. J. High Perform. Comput. 14 (2000) 102–136.

[58] N. Peters, J. Warnatz (Eds.), Numerical Methods in Laminar Flame Propagation, Friedr Vieweg und Sohn, Braunschweig, 1982.

[59] M.D. Smooke, M.L. Koszykowski, Fully adaptive solutions of one-dimensional mixed initial-boundary value problems with applications to unstable problems in combustion, SIAM J. Sci. Stat. Comput. 7 (1986) 301–321.

[60] A. Bayliss, B.J. Matkowsky, Fronts, relaxation oscillations, and period doubling in solid fuel combustion, J. Comput. Phys. 71 (1987) 147–168.