INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **57**:1771–1792
Published online 8 December 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1683

The locally conservative Galerkin (LCG) method for solving the incompressible Navier–Stokes equations

C. G. Thomas, P. Nithiarasu*, † and R. L. T. Bevan

Civil and Computational Engineering Centre, School of Engineering, Swansea University, Swansea SA2 8PP, U.K.

SUMMARY

In this paper, the locally conservative Galerkin (LCG) method (*Numer. Heat Transfer B Fundam.* 2004; **46**:357–370; *Int. J. Numer. Methods Eng.* 2007) has been extended to solve the incompressible Navier–Stokes equations. A new correction term is also incorporated to make the formulation to give identical results to that of the continuous Galerkin (CG) method. In addition to ensuring element-by-element conservation, the method also allows solution of the governing equations over individual elements, independent of the neighbouring elements. This is achieved within the CG framework by breaking the domain into elemental sub-domains. Although this allows discontinuous trial function field, we have carried out the formulation using the continuous trial function space as the basis. Thus, the changes in the existing CFD codes are kept to a minimum. The edge fluxes, establishing the continuity between neighbouring elements, are calculated via a post-processing step during the time-stepping operation. Therefore, the employed formulation needs to be carried out using either a time-stepping or an equivalent iterative scheme that allows post-processing of fluxes. The time-stepping algorithm employed in this paper is based on the characteristic-based split (CBS) scheme. Both steady- and unsteady-state examples presented show that the element-by-element formulation employed is accurate and robust.

Received 19 January 2007; Revised 1 October 2007; Accepted 14 October 2007

KEY WORDS: explicit local conservation; element-by-element solution; LCG; CBS; incompressible flow

1. INTRODUCTION

The conservation of the finite element method has been the topic of discussion for many investigators over the last several years [1–6]. Although the standard Galerkin form is implicitly locally conservative [5], its explicit application was put to practice only recently [7–9]. Explicit use of

*Correspondence to: P. Nithiarasu, Civil and Computational Engineering Centre, School of Engineering, Swansea University, Swansea SA2 8PP, U.K.
†E-mail: P.Nithiarasu@swansea.ac.uk

Contract/grant sponsor: EPSRC; contract/grant number: E/D070554/01

the element-wise conservation properties gives control over individual elements and their boundaries. Such control has several advantages in computational mechanics. For example, control over individual elements allows local problems to be solved, and thus it can be a useful tool for multiscale or multiresolution mechanics calculations. Control over element interfaces or edges gives the flexibility of introducing interface discontinuities. Also, such a control will be extremely useful in difficult problems of interest such as conjugate heat transfer [10], porous medium–fluid interface [11] and fluid–structure interaction [12].

Over the last 10 years, the Discontinuous Galerkin Method (DGM) [13–25] has been developed to remedy some of the drawbacks of the continuous (or global) Galerkin method. The DGM naturally allows element-by-element solution and thus can be easily paralised. The element-by-element solution, independent of the surrounding elements, allows independent, element-based change in the order of interpolation polynomials. The DGM is naturally element-wise conservative, and discontinuity capturing at domain interfaces is automatic. The DGM method can be highly accurate due to the above properties. However, these favourable properties come with the penalty of excessive number of degrees of freedoms and therefore more expensive than the continuous Galerkin (CG) (global) method. Thus, it is not surprising that researchers are looking for a DGM with a structure of CG (global) method [6]. We have adopted an approach, in which we can reintroduce the interface fluxes within the CG (global) framework [7, 8]. This way existing industrial fluid dynamic codes can be modified to accommodate the changes without resorting to complete revision of the codes. The reintroduction of the edge fluxes not necessarily has to break the trail function spaces but can be constructed within the continuous framework. However, via a post-processing approach the solution can be sought element-by-element. Such a method is identical to the CG or stabilized CG (global) methods under certain conditions [7–9] such as steady state. Away from such conditions, results deviate from that of GG results, especially when both convection and diffusion are used and transient solutions are sought. Thus, in addition to the standard locally conservative Galerkin (LCG) introduced in the past [7–9], we have also introduced a correction term to obtain identical results to that of GG method here. The LCG method used here is both locally and globally conservative and maintains many of the advantages that DGM offers, but at a lower cost. The only constraint of the method is that some form of iterative procedure (such as time stepping) is essential to obtain a post-processed flux after every iteration. This is not a great disadvantage, as some form of iterative procedure is essential for solving the N–S equations with many other type of solution strategies.

In the LCG method presented here, a time-stepping algorithm is coupled with a post-processing stage, in which all the necessary fluxes are computed from the previous time-step solution. Thus, the modifications needed for any existing GG codes in space–time frame are little. The changes needed for solving the system, element-by-element, are also straightforward. Since the standard finite element assembly is completely eliminated in the LCG procedure, the elemental matrices may be computed, inverted and stored at the pre-processing stages of a computation. Often these matrices are functions of the element property, such as area. Such properties normally do not change in an Eulerian frame of reference, and thus, control over individual elements gives a great deal of freedom to optimize memory requirement. Majority of the LCG forms, especially the implicit discretization, are much faster than their CG (or global) counterparts [7, 8]. In our previous works, we have demonstrated the LCG method's equivalence to the global Galerkin method [7, 8]. It was also demonstrated that how stabilized LCG forms can be constructed and employed. However, the problems of interest in the previous articles were pure diffusion and convection–diffusion, and the suitability of the LCG method for the solution of the Navier–Stokes equations was not

DOI: 10.1002/fld

considered. In the present work, we address the extension of the LCG method to the Navier–Stokes equations and also discuss the implementation in detail. We also introduce an important change to the standard LCG formulation to obtain identical solution to that of stabilized CG methods.

The paper is organized as follows. In Section 2, the governing equations and their temporal discretization, using the characteristic-based split (CBS) scheme, are briefly introduced. The LCG spatial discretization of the semi-discrete equations are discussed in Section 3. The discussion in Section 3 includes matrices arising from the spatial discretization and edge-flux calculation, and it also introduces a brief summary of local and dual time-stepping procedures. Numerical examples are provided in Section 4 to demonstrate the validity and accuracy of the LCG method for incompressible flows. These examples consist of the classic steady-state problems of lid-driven cavity flow and unsteady flow past a circular cylinder. All the results obtained are thoroughly analysed and compared with the CG (global) method and other available results wherever possible. Finally, in Section 5, some important conclusions are derived.

2. GOVERNING EQUATIONS AND THE CBS SCHEME

2.1. The Navier–Stokes equations for incompressible flow

The artificial compressibility-based Navier–Stokes equations may be expressed as

Continuity:

$$\frac{1}{\beta^2}\frac{\partial p}{\partial t}+\rho\frac{\partial u_i}{\partial x_i}=0 \tag{1}$$

Momentum:

$$\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}+\frac{1}{\rho}\frac{\partial p}{\partial x_i}-\frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j}=0 \tag{2}$$

where u_i are the cartesian components of the velocity vector, ρ is the fluid density, p represents the pressure and β is an artificial compressibility parameter [26–32]. The deviatoric stress components τ_{ij} are related to velocity gradients by

$$\tau_{ij}=\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{3}$$

where μ is the dynamic viscosity. The non-dimensional form of Equations (1) and (2) are rewritten as

Continuity:

$$\frac{1}{\beta^{*2}}\frac{\partial p^*}{\partial t^*}=-\frac{\partial u_i^*}{\partial x_i^*} \tag{4}$$

Momentum:

$$\frac{\partial u_i^*}{\partial t^*}=-u_j^*\frac{\partial u_i^*}{\partial x_j^*}+\frac{1}{Re}\frac{\partial^2 u_i^*}{\partial x_j^{*2}}-\frac{\partial p^*}{\partial x_i^*} \tag{5}$$

DOI: 10.1002/fld

where

$$x_i^* = \frac{x_i}{L}, \quad u_i^* = \frac{u_i}{u_\infty}, \quad t^* = \frac{t u_\infty}{L}, \quad p^* = \frac{p}{\rho u_\infty^2}, \quad \beta^{*2} = \frac{\beta^2}{u_\infty^2}, \quad Re = \frac{\rho_\infty u_\infty L}{\mu_\infty} \tag{6}$$

Here, Re is the Reynolds number, the subscript '∞' indicates a reference quantity and L is a reference length. The problem definition is completed by selecting appropriate initial and boundary conditions.

2.2. Semi-discrete form of the CBS scheme

The CBS scheme is a well-established algorithm in the CG (or global) context for both incompressible and compressible flow simulations [29, 33, 34]. The artificial compressibility form has been incorporated into the CBS family in 2003 and has been successfully used since [26, 27, 32, 35, 36]. For a recent, up to date account on the progress of the CBS scheme, the reader is referred to [31]. The analysis of the artificial compressibility form may be found in [32].

The present work couples the semi-discrete CBS scheme with the LCG spatial discretization procedure to obtain an element-wise solution strategy. The CBS approach adopted in this work, starts with a solution to an auxiliary intermediate velocity field. This intermediate velocity field is corrected, once the pressure field is obtained from a pressure (continuity) equation. By dropping the pressure term, Equation (5) becomes (asterisks are dropped for simplicity)

$$\frac{\partial u_i^\dagger}{\partial t} = -u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{Re} \frac{\partial^2 u_i}{\partial {x_j}^2} \tag{7}$$

where '$\dagger$' indicates the intermediate velocity field. To incorporate the LCG spatial discretization, Equation (7) is rewritten as

$$\frac{\partial u_i^\dagger}{\partial t} + \frac{\partial F_{ij}}{\partial x_j} = 0 \tag{8}$$

where the flux term, F_{ij}, is defined as

$$F_{ij} = \left(u_j u_i - \frac{1}{Re} \frac{\partial u_i}{\partial x_j} \right) \tag{9}$$

Applying the *simple explicit* characteristic temporal discretization [37] to Equation (8) gives

$$\frac{u_i^\dagger - u_i^n}{\Delta t} = -\left(\frac{\partial F_{ij}}{\partial x_j} \right)^n + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial F_{ij}}{\partial x_j} \right)^n \tag{10}$$

This forms the first step of the LCG-CBS scheme. The correct value of u_i^{n+1} is given by

$$u_i^{n+1} = u_i^\dagger - \Delta t \left(\frac{\partial p}{\partial x_i} \right)^n + \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k} \left(\frac{\partial p}{\partial x_i} \right)^n \tag{11}$$

DOI: 10.1002/fld

which is the third step of the scheme. Before the velocity correction can be applied, the pressure has to be calculated independently from another source in the second step. The pressure equation used is based on the continuity relation given by Equation (4)

$$\frac{1}{\beta^2}\frac{\Delta p}{\Delta t} = -\rho\frac{\partial u_i^{n+1}}{\partial x_i} \tag{12}$$

The u_i^{n+1} term appearing in this equation has to be eliminated, as it is unknown. Using Equation (11) and neglecting terms higher than second order, the pressure equation becomes

$$\frac{1}{\beta^2}\frac{\Delta p}{\Delta t} = \frac{1}{\beta^2}\frac{p^{n+1}-p^n}{\Delta t} = -\rho\frac{\partial}{\partial x_i}\left(u_i^\dagger - \Delta t\left(\frac{\partial p}{\partial x_i}\right)^n\right) \tag{13}$$

Equation (13) is the second step of the scheme. It has been derived by assuming the presence of a pseudo-compressibility in the continuity equation. This way it is possible to keep the fully explicit nature of the scheme. An artificial compressibility parameter, β, has been successfully used in the literature [26–32, 35, 38]. The value of β can be given as a constant throughout the domain, but the recommended approach, which is adopted here, is to calculate β locally, based on both convective and diffusive time-step restrictions [26, 27, 29]. This gives not only a scheme that is suitable for different Reynolds numbers but also more importantly it accommodates different flow regimes (convection and diffusion dominated) within a problem at a particular Reynolds number. In this work, the relation

$$\beta = \max(\varepsilon, v_{\text{conv}}, v_{\text{diff}}) \tag{14}$$

is employed. The constant ε, appearing in Equation (14), ensures that β does not approach zero and typically takes the value of $0.1 \leqslant \varepsilon \leqslant 0.5$. v_{conv} is the local convective velocity and v_{diff} is the local diffusive velocity. These velocities are calculated from the non-dimensional relations [26],

$$v_{\text{conv}} = \sqrt{u_i u_i}, \quad v_{\text{diff}} = \frac{1}{h\,Re} \tag{15}$$

The three steps of the LCG-CBS scheme may be summarized as:

1. Solve for the intermediate velocity field, $u_i^\dagger$, using Equation (10).
2. Solve for the pressure field, p, using Equation (13).
3. Solve for the correct velocity field, u_i, using Equation (11).

3. LCG SPATIAL DISCRETIZATION

In this section, Equations (10), (13) and (11) are discretized in space using the LCG finite element procedure [7–9]. As with the global Galerkin form, the variation of each of the variables is approximated by the standard finite element spatial discretization as

$$u_i \approx \tilde{u}_i = \mathbf{N}\mathbf{u_i} \quad \text{and} \quad p \approx \tilde{p} = \mathbf{N}\mathbf{p} \tag{16}$$

where $\mathbf{N}$ are the shape functions. The CBS scheme adopted circumvents the LBB condition when the pressure gradient is removed from the momentum equation [29, 33]. This allows an arbitrary

DOI: 10.1002/fld

choice of shape functions to be made for approximating the velocity and pressure fields. Here, all variable approximations use identical shape functions. The optimal choice of weighting function for the semi-discrete equations is the Galerkin weighting, $w_a = N_a$, when using a CBS discretization [29]. In an LCG discretization, the need for breaking the trial function spaces never arises unless such breaking of spaces is needed. However, the domain can be broken into elemental subdomains, thanks to the post-processed flux. To make the presentation brief, we provide only the discretization over broken elemental subdomains.

With the Galerkin weighting, the integral form of the first step may be expressed as

$$\int_{\Omega_e} N_a \frac{\Delta \tilde{u}_i^{\dagger}}{\Delta t}\,\mathrm{d}\Omega_e = -\int_{\Omega_e} N_a \left(\frac{\partial \tilde{F}_{ij}}{\partial x_j}\right)^n \mathrm{d}\Omega_e + \int_{\Omega_e} N_a \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial \tilde{F}_{ij}}{\partial x_j}\right)^n \mathrm{d}\Omega_e \tag{17}$$

Integration by parts gives the following weak form:

$$\begin{aligned}\int_{\Omega_e} N_a \frac{\Delta \tilde{u}_i^{\dagger}}{\Delta t}\,\mathrm{d}\Omega_e = {} & \int_{\Omega_e} \frac{\partial N_a}{\partial x_j} \tilde{F}_{ij}^n \,\mathrm{d}\Omega_e - \int_{\Gamma_e} N_a \hat{\tilde{F}}_{ij}^n \,\mathrm{d}\Gamma_e n_j \\ & - \int_{\Omega_e} \left(\frac{\Delta t}{2} u_k \frac{\partial N_a}{\partial x_k}\right)\left(\frac{\partial \tilde{F}_{ij}}{\partial x_j}\right)^n \mathrm{d}\Omega_e \\ & + \int_{\Gamma_e} \left(\frac{\Delta t}{2} u_k N_a\right)\left(\frac{\partial \hat{\tilde{F}}_{ij}}{\partial x_j}\right)^n \mathrm{d}\Gamma_e n_k \end{aligned} \tag{18}$$

In the LCG method, the boundary terms are replaced with a numerical flux, $\hat{\tilde{F}}_{ij}$ and $\partial \hat{\tilde{F}}_{ij}/\partial x_j$, calculated at time (n) on the element boundaries. Neglecting third- and higher-order terms, the final matrix form of the first step is

$$[\mathbf{M1_e}]\{\Delta \mathbf{u_i^{\dagger}}\} = \Delta t\,([\mathbf{K1_e}]\{\mathbf{u_i}\} + [\mathbf{K1_e^{cg}}]\{\mathbf{u_i}\} + \{\mathbf{f\hat{1}_e}\})^n \tag{19}$$

$$\begin{aligned}[\mathbf{M1_e}] &= \int_{\Omega_e} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Omega_e, \quad [\mathbf{K1_e}] = \int_{\Omega_e} \frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_j}\left(u_j \mathbf{N} - \frac{1}{Re}\frac{\partial \mathbf{N}}{\partial x_j}\right)\mathrm{d}\Omega_e \\ [\mathbf{K1_e^{cg}}] &= \int_{\Omega_e} \left(\frac{\Delta t}{2} u_k\right)\frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_k} u_j \frac{\partial \mathbf{N}}{\partial x_j}\,\mathrm{d}\Omega_e \\ \{\mathbf{f\hat{1}_e}\}^n &= -\int_{\Gamma_e} \mathbf{N}^{\mathrm{T}} u_j \mathbf{N}\,\mathrm{d}\Gamma_e n_j \{\mathbf{u_i}\}^n + \frac{1}{Re}\int_{\Gamma_e} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Gamma_e n_j \left\{\frac{\partial \hat{\mathbf{u}}_\mathbf{i}}{\partial x_j}\right\}^n \\ &\quad + \int_{\Gamma_e} \left(\frac{\Delta t}{2} u_k\right)\mathbf{N}^{\mathrm{T}} u_j \mathbf{N}\,\mathrm{d}\Gamma_e n_k \left\{\frac{\partial \hat{\mathbf{u}}_\mathbf{i}}{\partial x_j}\right\}^n \end{aligned} \tag{20}$$

In a similar fashion, the weighted residual form of the second step may be expressed as

$$\int_{\Omega_e} N_a \left(\frac{1}{\beta^2}\right)^n \frac{\Delta \tilde{p}}{\Delta t}\,\mathrm{d}\Omega_e = -\rho \int_{\Omega_e} N_a \frac{\partial}{\partial x_i}\left(\tilde{u}_i^{\dagger} - \Delta t \left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n\right)\mathrm{d}\Omega_e \tag{21}$$

DOI: 10.1002/fld

and the weak form is

$$\int_{\Omega_e} N_a \left(\frac{1}{\beta^2}\right)^n \frac{\Delta\tilde{p}}{\Delta t}\,\mathrm{d}\Omega_e$$

$$= \rho \int_{\Omega_e} \frac{\partial N_a}{\partial x_i}\left(\tilde{u}_i^{\dagger} - \Delta t\left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n\right)\mathrm{d}\Omega_e - \rho \int_{\Gamma_e} N_a \left(\tilde{u}_i^{\dagger} - \Delta t\left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n\right)\mathrm{d}\Gamma_e n_i \tag{22}$$

The final matrix form of the second step is

$$[\mathbf{M2_e}]\{\Delta\mathbf{p}\} = \Delta t\left([\mathbf{C2_e}]\{\mathbf{u_i^{\dagger}}\} - \Delta t[\mathbf{K2_e}]\{\mathbf{p}\}^n + \{\hat{\mathbf{f}}\mathbf{2_e}\}\right) \tag{23}$$

The matrices used in the above equation are defined as

$$[\mathbf{M2_e}] = \int_{\Omega_e} \mathbf{N}^{\mathrm{T}}\left(\frac{1}{\beta^2}\right)^n \mathbf{N}\,\mathrm{d}\Omega_e, \quad [\mathbf{C2_e}] = \int_{\Omega_e} \frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_i}\mathbf{N}\,\mathrm{d}\Omega_e$$

$$[\mathbf{K2_e}] = \int_{\Omega_e} \frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_i}\frac{\partial \mathbf{N}}{\partial x_i}\,\mathrm{d}\Omega_e \tag{24}$$

$$\{\hat{\mathbf{f}}\mathbf{2_e}\}^n = -\int_{\Gamma_e} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Gamma_e n_i \left(\{\mathbf{u_i^{\dagger}}\} - \Delta t\left\{\frac{\partial \hat{\mathbf{p}}}{\partial x_i}\right\}^n\right)$$

The weighted residual and weak forms of the third step are

$$\int_{\Omega_e} N_a \tilde{u}_i^{n+1}\,\mathrm{d}\Omega_e = \int_{\Omega_e} N_a \tilde{u}_i^{\dagger}\,\mathrm{d}\Omega_e - \Delta t \int_{\Omega_e} N_a \left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n \mathrm{d}\Omega_e$$

$$+ \int_{\Omega_e} N_a \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k}\left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n \mathrm{d}\Omega_e \tag{25}$$

and

$$\int_{\Omega_e} N_a \tilde{u}_i^{n+1}\,\mathrm{d}\Omega_e = \int_{\Omega_e} N_a \tilde{u}_i^{\dagger}\,\mathrm{d}\Omega_e + \Delta t \int_{\Omega_e} \frac{\partial N_a}{\partial x_i}\tilde{p}^n\,\mathrm{d}\Omega_e - \Delta t \int_{\Gamma_e} N_a \hat{\tilde{p}}^n\,\mathrm{d}\Gamma_e n_i$$

$$- \int_{\Omega_e} \frac{\Delta t^2}{2} u_k \frac{\partial N_a}{\partial x_k}\left(\frac{\partial \tilde{p}}{\partial x_i}\right)^n \mathrm{d}\Omega_e + \int_{\Gamma_e} \frac{\Delta t^2}{2} u_k N_a \left(\frac{\partial \hat{\tilde{p}}}{\partial x_i}\right)^n \mathrm{d}\Gamma_e n_k \tag{26}$$

The final matrix form of the third step is

$$[\mathbf{M3_e}]\{\mathbf{u_i}\}^{n+1} = [\mathbf{M3_e}]\{\mathbf{u_i}\}^{\dagger} + \Delta t\left([\mathbf{K3_e}]\{\mathbf{p}\} - [\mathbf{K3_e^{cg}}]\{\mathbf{p}\} + \{\mathbf{f3_e}\}\right)^n \tag{27}$$

DOI: 10.1002/fld

The matrices used in the above equation are defined as

$$[\mathbf{M3_e}] = \int_{\Omega_e} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Omega_e, \quad [\mathbf{K3_e}] = \int_{\Omega_e} \frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_i}\mathbf{N}\,\mathrm{d}\Omega_e$$
$$[\mathbf{K3_e^{cg}}] = \int_{\Omega_e} \left(\frac{\Delta t}{2}u_k\right)\frac{\partial \mathbf{N}^{\mathrm{T}}}{\partial x_k}\frac{\partial \mathbf{N}}{\partial x_i}\,\mathrm{d}\Omega_e \tag{28}$$
$$\{\mathbf{f3_e}\}^n = -\int_{\Gamma_e} \mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Gamma_e n_i \{\mathbf{p}\}^n + \int_{\Gamma_e}\left(\frac{\Delta t}{2}u_k\right)\mathbf{N}^{\mathrm{T}}\mathbf{N}\,\mathrm{d}\Gamma_e n_k \left\{\frac{\hat{\partial}\mathbf{p}}{\partial x_i}\right\}^n$$

The third- and higher-order terms are neglected from all the equations for simplicity.

3.1. Calculation of edge fluxes

To satisfy local conservation, the flux crossing a common edge shared by two elements is made equal and forced to act in opposing directions as shown in Figure 1. Using the computed edge flux on the element boundaries of adjoining elements, the following condition is enforced:

$$F_{e1}n_{e1} + F_{e2}n_{e2} = 0 \tag{29}$$

where $F_{e1} = F_{e2}$, and n_{e1} and n_{e2} are the outward normals from the edges of the respective elements. The subscripts are defined in Figure 1.

All that is needed now to enforce the condition given by Equation (29) is a small post-processing calculation to be made at the end of each time-step. This is possible once a globally continuous solution is recovered from the previous time-step. The post-processing calculation uses the nodal values of the continuous solution and its gradients to provide an accurate interface flux. It is this flux that establishes connectivity between elements at the next time-step.

For linear triangular finite elements, the gradient of the variable is a constant within each element. The standard procedure to obtain a nodal approximation of the variable's gradient, when using linear elements [7–9, 39], is to compute a mean average of the constant gradients over the

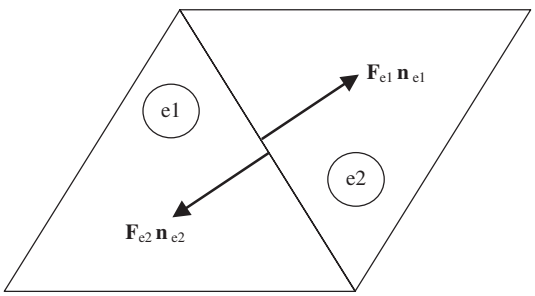


Figure 1. Flux crossing a common edge between two elements.

DOI: 10.1002/fld

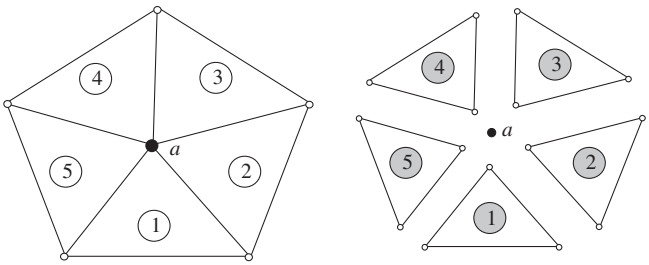


Figure 2. A patch of linear elements. Element-averaged gradient calculation of the scalar variable at node a.

elements surrounding a node of interest. For example, a gradient value of a scalar variable ϕ, at node a in Figure 2, may be computed as

$$\left(\frac{\hat{\partial}\phi}{\partial x_i}\right)_a = \frac{1}{\text{ne}} \sum_{e=1}^{\text{ne}} \left(\frac{\partial \phi}{\partial x_i}\right)_e \tag{30}$$

These nodal gradient values are then used to estimate the diffusive component of flux crossing the edges as an average of the two nodal values forming an edge.

3.2. Correction factor for the LCG formulation

In principle, the equal and opposite fluxes introduced at the interfaces of the elements will cancel each other when the solution is averaged over a node for a steady-state problem. However, for transient problems, a small difference in the flux terms due to the mass matrix may exist on non-uniform meshes. This needs to be corrected by adding the difference between the opposite fluxes on the element edges.

The only other difference between the GG and the LCG equations described here is the differences in the use of the mass matrix. In order for the LCG method to produce the same results to that of the CG method, a correction factor based on this difference must be implemented. Without this correction, the formulation may not give the results identical to that of the GG formulation. This section details the correction factor utilized for a case where a lumped mass matrix is employed.

The LCG elemental equation for step 1 is repeated for clarity from (19):

$$\{\mathbf{u_i}^{\dagger}\} = \{\mathbf{u_i^n}\} + \Delta t[\mathbf{M1_e}]^{-1}([\mathbf{K1_e}]\{\mathbf{u_i}\} + [\mathbf{K1_e^{cg}}]\{\mathbf{u_i}\} + \{\hat{\mathbf{f}}\mathbf{1_e}\}) \tag{31}$$

Examining a nodal equation and noting that the LCG method produces multiple solutions at a node and that an arithmetic mean is employed to produce a single solution, the equation for node a is

$$\{u_{ia}^{\dagger}\} = \{u_{ia}^{n}\} + \frac{1}{\text{ne}}\Delta t[\mathbf{M1_e}]^{-1}([\mathbf{K1_e}]\{\mathbf{u_i}\} + [\mathbf{K1_e^{cg}}]\{\mathbf{u_i}\} + \{\hat{\mathbf{f}}\mathbf{1_e}\}) \tag{32}$$

where ne is the number of elements connected to node a. Since $[\mathbf{M1_e}]$ is assumed to be lumped, this is not a matrix in the above equation but a number and it is a function of the element size. The corresponding CG method (interior node) is of the form

$$\{u_{ia}^{\dagger}\} = \{u_{ia}^{n}\} + \Delta t[\mathbf{M1}]^{-1}([\mathbf{K1}_e]\{\mathbf{u_i}\} + [\mathbf{K1_e^{cg}}]\{\mathbf{u_i}\}) \tag{33}$$

DOI: 10.1002/fld

where [**M1**] is the sum of the individual elemental lumped mass matrix contributions. Comparison between Equations (32) and (33) clearly shows that the difference is in the use of the mass matrix. Thus, a correction of the type

$$\left(\Delta t[\mathbf{M1}]^{-1} - \frac{1}{\text{ne}}\Delta t[\mathbf{M1_e}]^{-1}\right)([\mathbf{K1_e}]\{\mathbf{u_i}\} + [\mathbf{K1_e^{cg}}]\{\mathbf{u_i}\}) \tag{34}$$

is added to Equation (32) to obtain results identical to that of the GG formulation for inside nodes. For the global boundary nodes, the results of the two methods differ if Dirichlet boundary conditions are used. However, they are identical if Newmann conditions are employed. The above correction is especially important when transient problems are solved. The correction factors can be determined at a pre-processing stage for cases involving the Eulerian form of reference. The correction is identical for all the three steps if the explicit CBS scheme is employed.

3.3. Local time stepping

The local time-step at each node a for the artificial compressibility method is given by

$$\Delta t_a = \min\left(\frac{h_a}{\sqrt{u_i u_i} + \beta}, \frac{h^2 Re}{2}\right) \tag{35}$$

where h_a is the local element-size at node a. Generally, h_a is calculated in two dimensions as the minimum size of all the surrounding elements [29], e, i.e.

$$h_a = \min(2\text{area}/\text{opposite side length})_e \tag{36}$$

The artificial compressibility parameter β in Equation (35) is calculated from Equation (14). The inclusion of β in Equation (35) allows the artificial wave speed to be included into the local time-step limit. The calculated Δt_a is in practice multiplied by a non-zero safety factor below unity. The actual value of the safety factor depends on the problem being simulated and the mesh used.

To examine the convergence of the artificial compressibility LCG-CBS scheme, the pressure residual error of the solution is calculated using [26, 27]

$$\text{error} = \frac{1}{\text{nnode}} \sum_{i=1}^{\text{nnode}} \left[\frac{1}{\beta^2}\frac{p^{n+1} - p^n}{\Delta t}\right]^2 \tag{37}$$

where nnode is the total number of nodes in the mesh. The criterion for reaching steady state is that this error should be reduced to a value of 1×10^{-7} or less.

3.4. Recovering a transient solution via a dual time-stepping approach

For the solution of transient incompressible-flow problems, a dual time-stepping procedure is implemented. The method has been shown to be very successful for both pre-conditioned artificial-compressibility, finite volume [28, 38, 40] and the global Galerkin fully-explicit CBS schemes [26, 27, 29, 31, 32, 35].

To recover the true transient solution, a real-time term is added to the momentum equation. To remain consistent with the global Galerkin CBS scheme, a real-time term is added to the third

DOI: 10.1002/fld

step [26, 27, 31, 32, 35]. The addition of a true transient term leads to the following modified third step:

$$\{\mathbf{u_i}\}^{n+1} = \{\mathbf{u_i}\}^{\dagger} + [\mathbf{M3_e}]^{-1}\Delta t([\mathbf{K3_e}]\{\mathbf{p}\} - [\mathbf{K3_e^{cg}}]\{\mathbf{p}\} + \{\mathbf{f3_e}\})^n - [\mathbf{M3_e}]^{-1}\frac{\Delta t}{\Delta \tau}\{\Delta \mathbf{u_i}\}^{\tau} \tag{38}$$

where $\Delta\tau$ is the real time-step. When using the dual time-stepping scheme, local Δt becomes a (pseudo) time-step for an iterative mechanism. In order to obtain a second-order real-time accuracy, $\{\Delta \mathbf{u_i}\}^{\tau}$ is approximated with an implicit second-order backward-difference formula of the type

$$\{\mathbf{u_i}\}^{\tau} = \frac{3\{\mathbf{u_i}\}^{m+1} - 4\{\mathbf{u_i}\}^{m} + \{\mathbf{u_i}\}^{m-1}}{2} \tag{39}$$

In the above equation, $\{\mathbf{u_i}\}^{m+1}$ is the nth pseudo-time level value within the pseudo-time loop, $\{\mathbf{u_i}\}^{m}$ is the steady-state solution at the last real time-step and $\{\mathbf{u_i}\}^{m-1}$ is the steady-state solution at one real time-step before the last. Clearly, both the latter vectors need to be appropriately stored at the end of each real time-step. Owing to the implicit nature of the scheme, the real time-step size is unrestricted and only governed by the quality of the transient solution required. The pseudo-time-step, however, is locally calculated and subjected to the usual stability conditions.

4. NUMERICAL EXAMPLES

4.1. Incompressible flow inside a lid-driven square cavity

The first problem considered is the incompressible fluid flow in a lid-driven cavity. The problem is investigated for a range of Reynolds numbers including the case of Stokes flow ($Re = 0$). Qualitative results are given using linear elements and comparisons of the velocity profiles are made with well-known benchmark data [41].

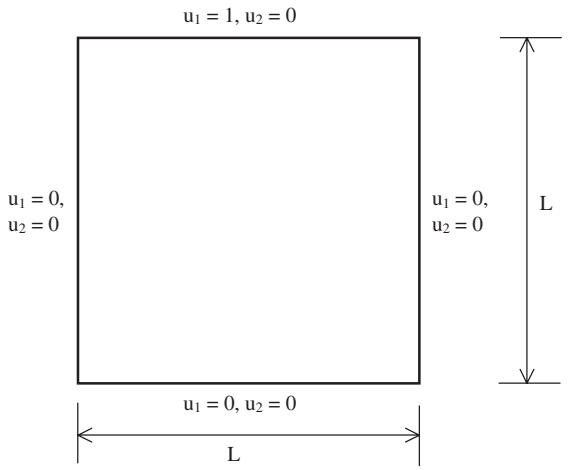


Figure 3. Flow in a square cavity. Geometry and boundary conditions.

DOI: 10.1002/fld

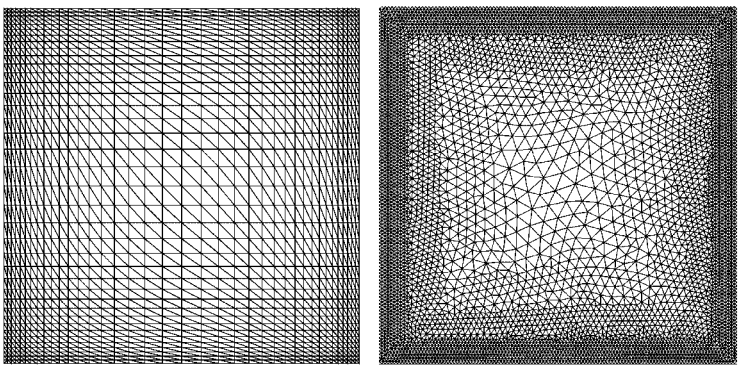


Figure 4. Flow in a square cavity. Meshes of linear elements were used in the computations: (a) Mesh B and (b) Mesh C.

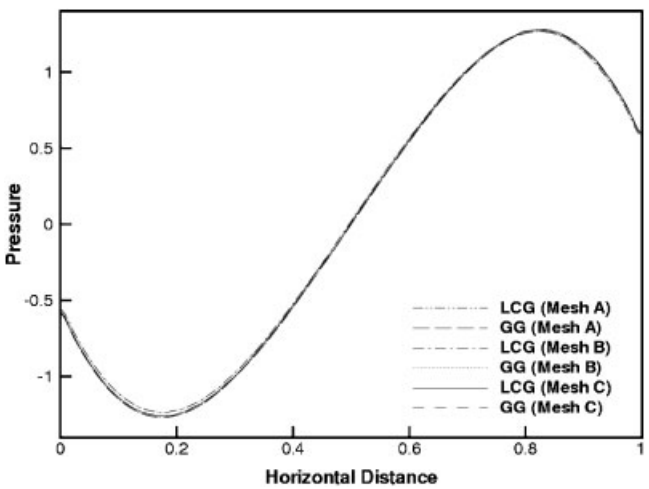


Figure 5. Stokes flow in a square cavity. Comparison of pressure distribution along mid-horizontal line.

The cavity geometry and boundary conditions are shown in Figure 3. No-slip boundary conditions are applied to the cavity walls and the flow inside the cavity is generated by the motion of the top surface, which travels in the horizontal direction. As the cavity is non-leaky, the singularities at the top two corners make this problem difficult to solve.

A number of meshes were used in this study, Mesh A is a uniform structured mesh of 20 000 linear triangular elements and 10 201 nodes. Mesh B is a non-uniform structured grid, with 1521 linear elements and 2888 nodes, and Mesh C is an unstructured mesh consisting of 10 596 elements and 5515 nodes. Meshes B and C are shown in Figure 4.

The case of Stokes flow is considered first, with all meshes being used in the study. To model Stokes flow, the convection and characteristic terms in Step 1 (Equation (10)) and the characteristic

DOI: 10.1002/fld

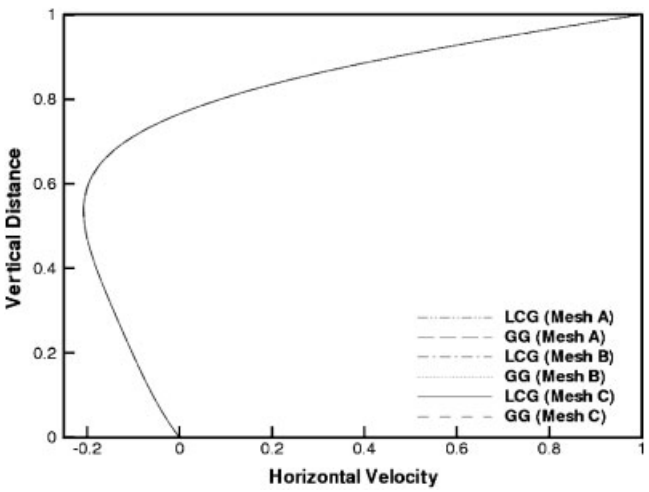


Figure 6. Stokes flow in a square cavity. Comparison of the horizontal velocity component along mid-vertical line.

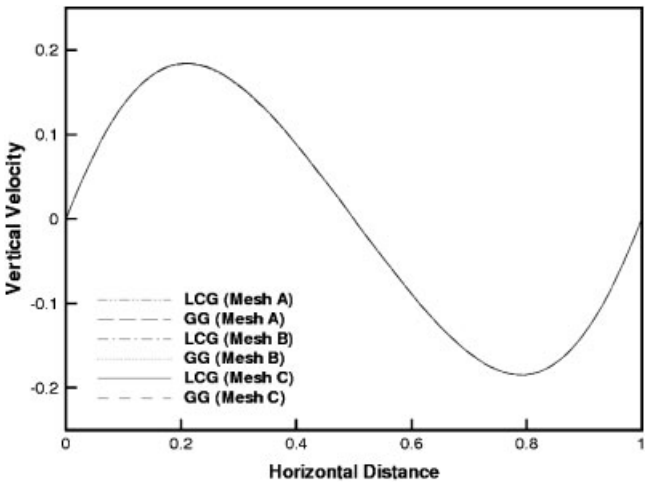


Figure 7. Stokes flow in a square cavity. Comparison of the vertical velocity component along mid-horizontal line.

term in Step 3 (Equation (11)) were switched off, also Re was set to unity and Step 2 was unaltered. The above changes give a matrix-free fractional time-stepping scheme, similar in structure to that described by Nithiarasu [35] for incompressible solid mechanics.

To provide a comparison of the accuracy of the pressure solution, the pressure along the mid-horizontal line has been plotted in Figure 5 for each mesh. The global Galerkin results are also plotted in this figure. As seen the agreement between the solutions obtained on Meshes A, B and

DOI: 10.1002/fld

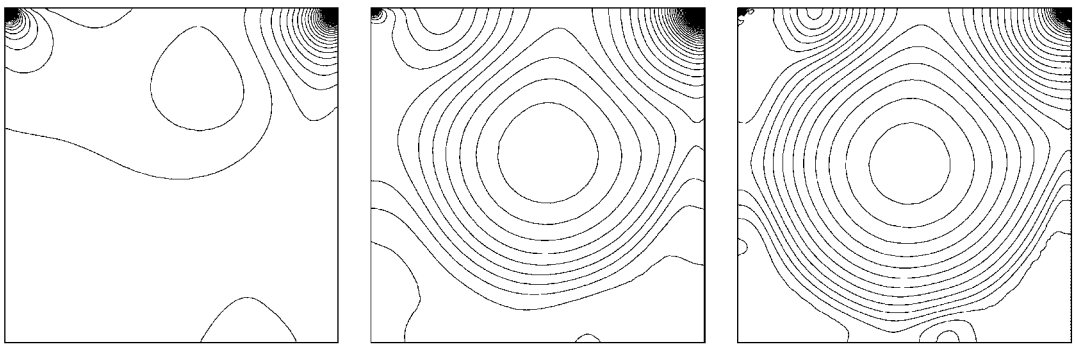


Figure 8. Flow in a lid-driven cavity. Pressure contours: (a) $Re=100$; (b) $Re=1000$; and (c) $Re=5000$.

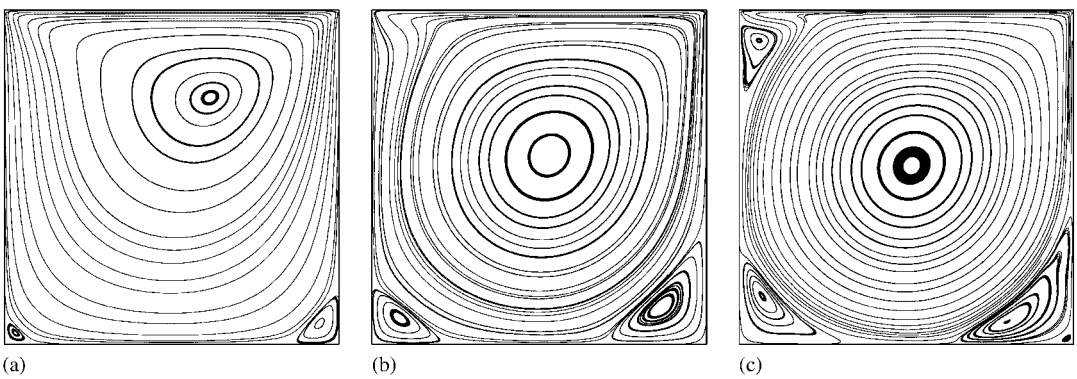


Figure 9. Flow in a lid-driven cavity. Streamtrace contours: (a) $Re=100$; (b) $Re=1000$; and (c) $Re=5000$.

C is excellent. The velocities plotted along the vertical and horizontal centrelines are shown in Figures 6 and 7. Again, the results show that the results on Meshes A, B, and C are in excellent agreement with each other and with the GG solution.

The lid-driven cavity problem is also solved at $Re=100, 400, 1000, 2000, 3200$ and 5000. The results for some selected Reynolds numbers are given in Figures 8–11. Pressure contours are given in Figure 8 and stream-trace plots are given in Figure 9 for different Reynolds numbers on the Mesh C. The contours are generally smooth and are without any major oscillations. Other meshes employed have also produced similar results.The results shown in Figures 8 and 9 are in very good agreement with GG solution and also with other methods. It is important to note that at $Re=5000$ the expected secondary recirculation at the bottom right corner is clearly predicted. This shows that the method presented is robust.

For a quantitative evaluation of the performance of the CBS-LCG scheme, the velocity components along the mid centrelines are compared with the benchmark solution by Ghia *et al.* [41]. The velocity distributions at various Reynolds numbers are given in Figures 10 and 11 for Mesh B.

DOI: 10.1002/fld

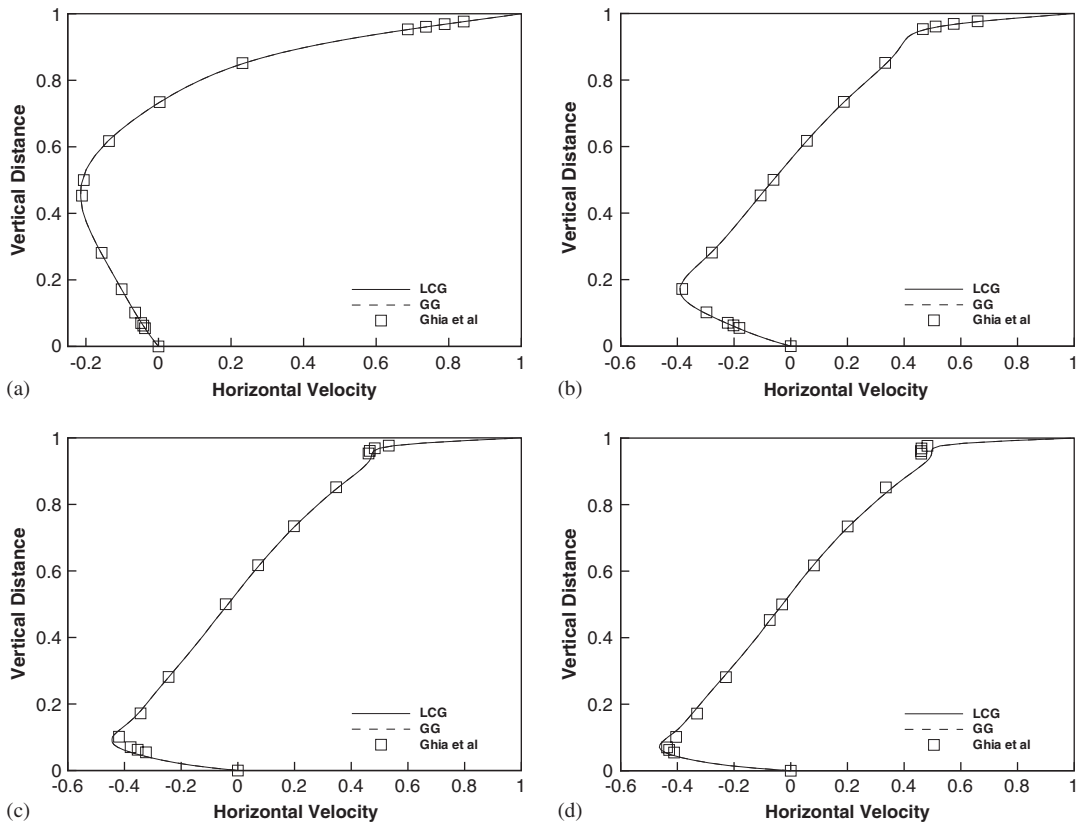


Figure 10. Flow in a lid-driven cavity. Comparison with Ghia *et al.* [41] of u_1 velocity distribution along the vertical centreline: (a) $Re=100$; (b) $Re=1000$; (c) $Re=3200$; and (d) $Re=5000$.

Overall, there is a good agreement between the LCG, GG methods and benchmark numerical solutions. The small deviation of the peak value at $Re=5000$ may be attributed to the degree of refinement used in the mesh.

4.2. Unsteady flow past a circular cylinder

In this subsection, the analysis is extended to investigate unsteady vortex shedding behind a circular cylinder at $Re=100$. As already discussed, the LCG-CBS scheme in its fully explicit nature is only suitable for steady-state solutions. However, the true transient solution is recovered here by using a dual time-stepping method [26–29, 31, 38, 40], which is incorporated into the LCG-CBS scheme.

The incompressible flow past a circular cylinder has been a popular test case for validating the transient part of numerical schemes [26, 27, 29, 32, 38, 42]. Figure 12 gives the problem statement and geometry. The computational domain is $16D$ in length and $8D$ in width. The centre of the cylinder is located at a distance of $4D$ from the inlet, along the centre line. At the inlet boundary,

DOI: 10.1002/fld

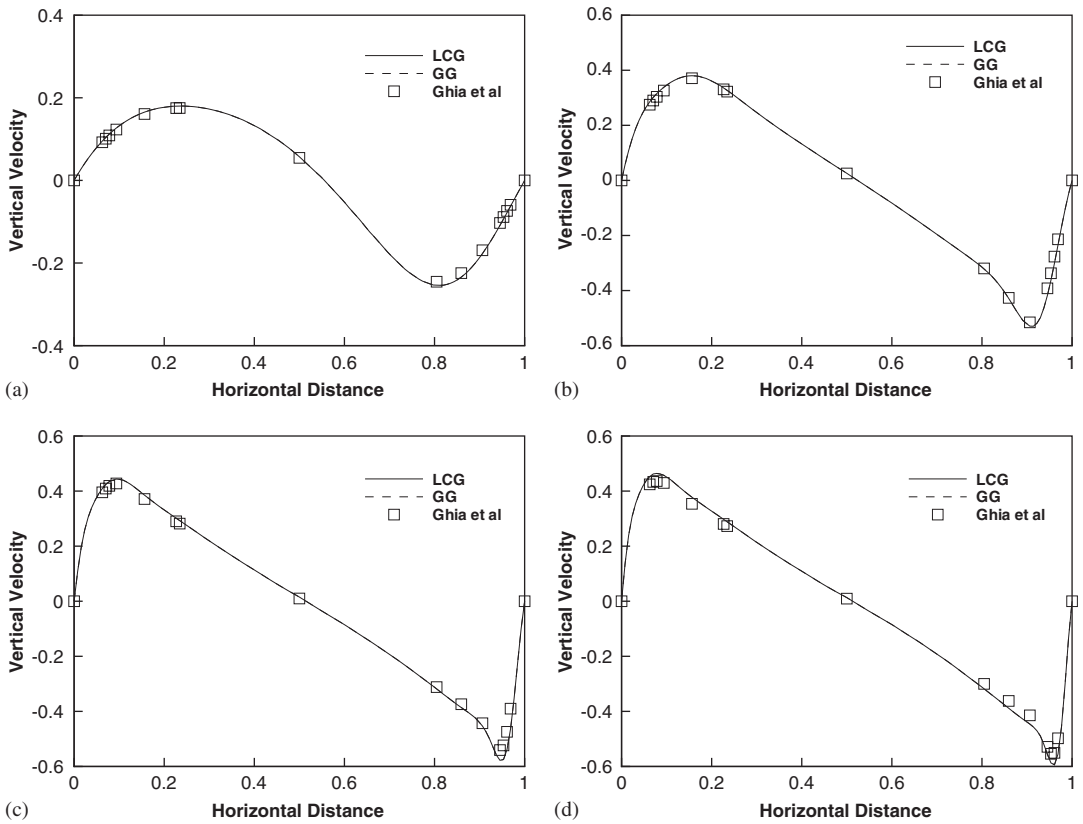


Figure 11. Flow in a lid-driven cavity. Comparison with Ghia *et al*. [41] of u_2 velocity distribution along the vertical centreline: (a) $Re=100$; (b) $Re=1000$; (c) $Re=3200$; and (d) $Re=5000$.

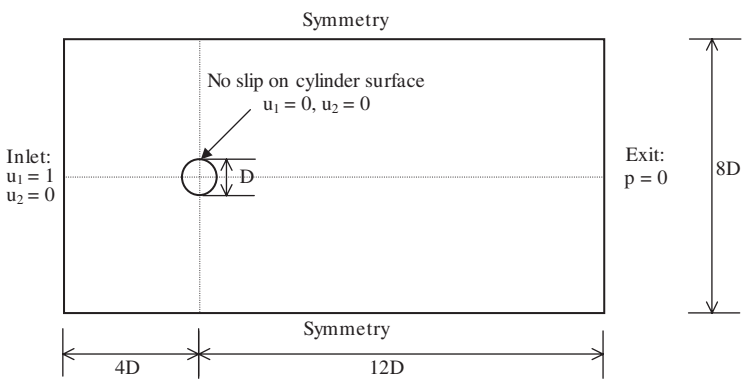


Figure 12. Unsteady flow past a circular cylinder. Geometry and boundary conditions.

DOI: 10.1002/fld

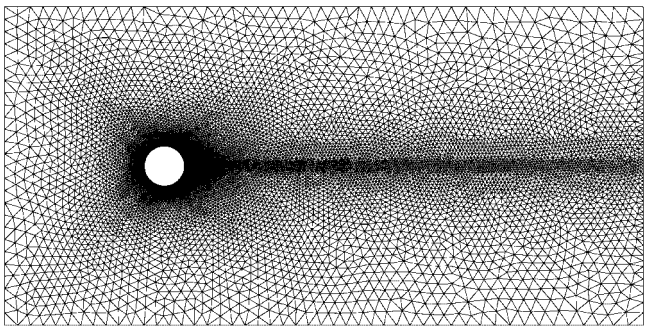


Figure 13. Unsteady flow past a circular cylinder at $Re=100$. Unstructured mesh used in the computations.

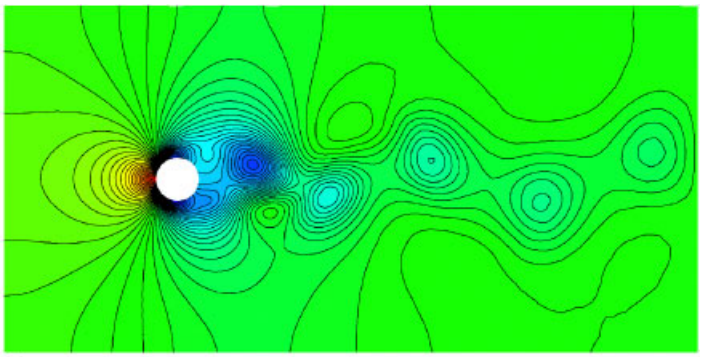
the horizontal and vertical velocity components are prescribed as unity and zero, respectively. At the exit, the pressure is set equal to zero. Slip conditions are imposed at the top and bottom extremities. The only solid surface in the domain is that of the cylinder and no-slip conditions are prescribed on it. At $t=0$, the initial conditions everywhere are: horizontal velocity is unity and both the vertical velocity and pressure are zero.

A single unstructured mesh was used in this study and it is shown in Figure 13. It has 19 650 elements and 9988 nodes. This mesh has been designed using previous knowledge on the formation of the unsteady wake and subsequent vortex shedding [43, 44]. The real time-step size chosen for this problem is 0.1 and simulations were carried out for a real non-dimensional time of 200. For each real time step, the pressure residual, given by Equation (37), was allowed to reach a tolerance of 1×10^{-7}.

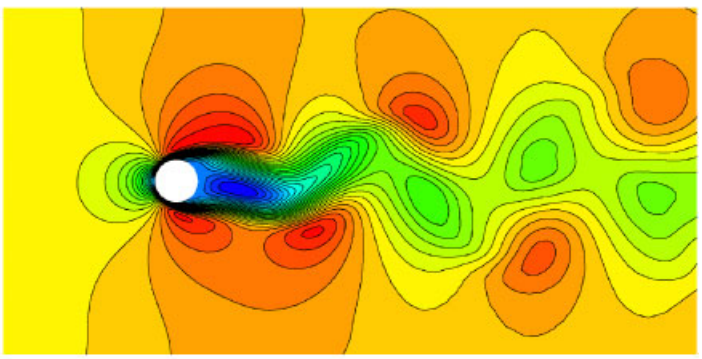
Some qualitative results are shown in Figure 14. Here, the contours of pressure and horizontal and vertical velocities are shown at the real non-dimensional time of 150. The results shown clearly indicate that the expected shedding patterns are obtained. These results are in close agreement with other qualitative results [26, 27, 38] and the GG method.

A quantitative analysis of the results was also conducted and is shown in Figure 15. Here, the real-time history of both the lift and drag coefficients are given along with the variation of the vertical velocity component at the central exit point. Figure 15(a) and (b) shows the coefficient of drag, c_d, computed over the cylinder surface as a function of time. The full temporal history of c_d for $1\leqslant t\leqslant200$ is given in Figure 15(a). Figure 15(b) shows a close-up view of the c_d for $130\leqslant t\leqslant150$. Here, the global Galerkin results have been included for comparison. As seen, the difference is negligible between the two results. Details of the coefficient of lift, c_l, are shown in Figure 15(c) and (d). The former gives the full real-time history and the latter gives a comparison with the global Galerkin results $130\leqslant t\leqslant150$. As seen the agreement here is excellent. The final set, Figure 15(e) and (f), gives the real-time history of the vertical velocity component—computed at the centre-point on the exit boundary. In Figure 15(f), the comparison with the global Galerkin results again shows an excellent agreement. The average drag coefficient obtained here is approximately 1.5. Previous studies, on a finer mesh, give an average drag coefficient value of 1.528 using orthogonal subscale stabilization (OSS) [45] and 1.521 using the second order CBS scheme [32]. This again shows that the results obtained using LCG method with the additional correction are in excellent agreement with other results.

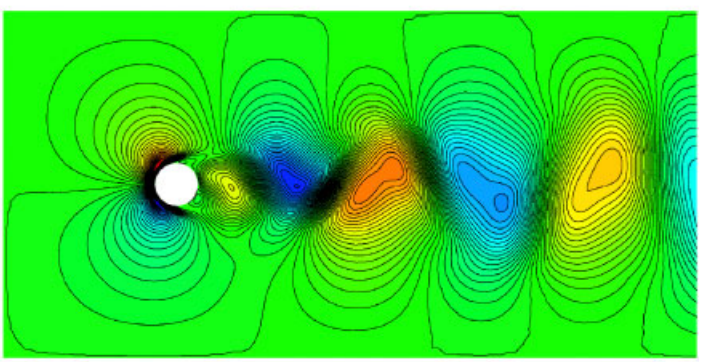
DOI: 10.1002/fld



(a)



(b)



(c)

Figure 14. Unsteady flow past a circular cylinder at $Re = 100$. Computed solution at a non-dimensional real time of 150. (a) Pressure contours, (b) u_1 velocity contours, and (c) u_2 velocity contours.

DOI: 10.1002/fld

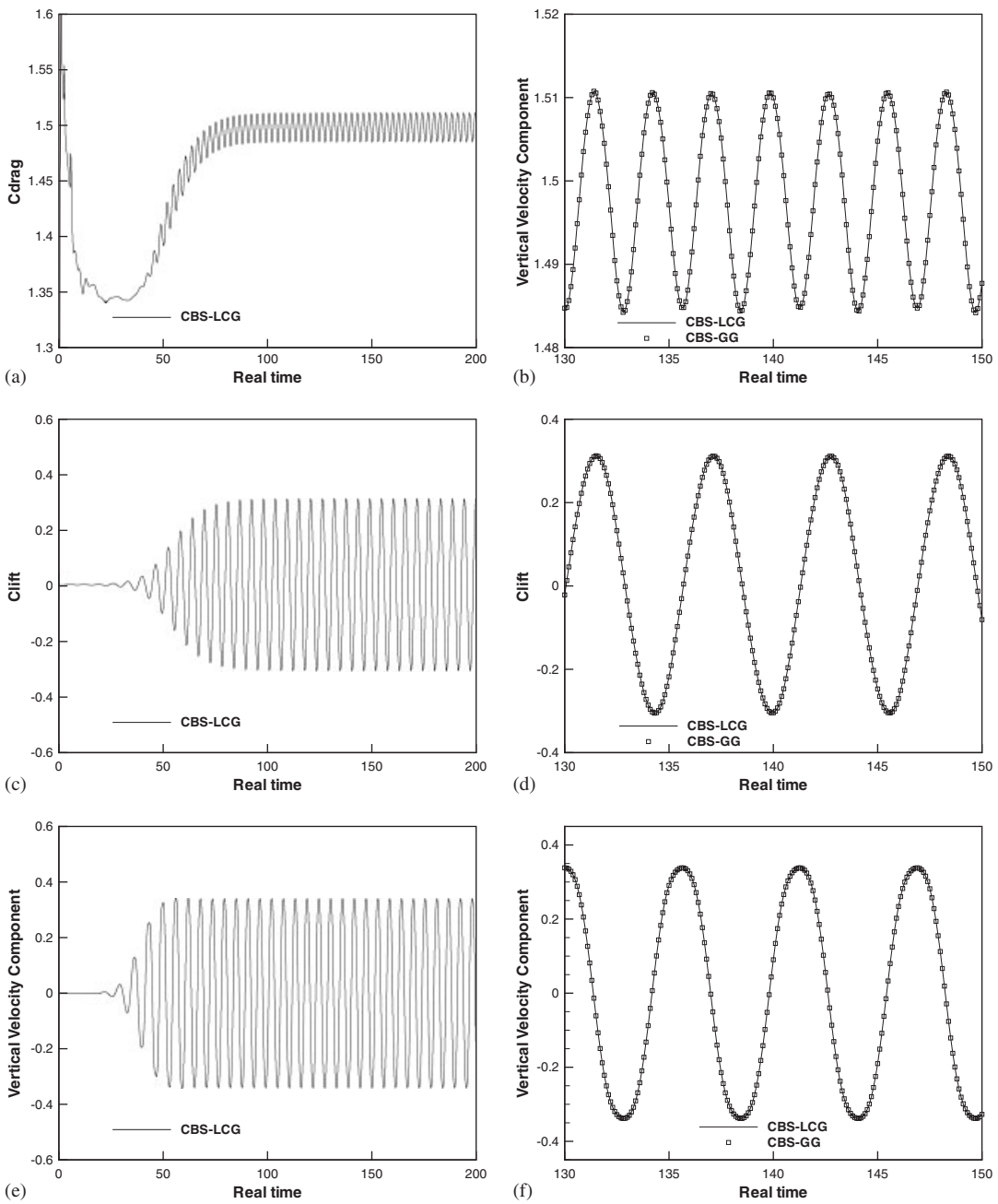


Figure 15. Unsteady flow past a circular cylinder at $Re = 100$. Computed coefficients of the drag and lift, and computed vertical velocity component at central exit point. All are plotted as a function of the non-dimensional real time. (a) Full history of drag coefficient c_d, (b) comparison of c_d with GG, (c) full history of lift coefficient c_l, (d) comparison of c_l with GG, (e) vertical velocity component u_2 at central exit, (f) comparison of u_2 at exit, with GG.

DOI: 10.1002/fld

5. CONCLUSIONS

The objective of this paper was to implement and demonstrate the locally conservative Galerkin (LCG) method for incompressible flows. This has been successfully achieved. In addition to implementation of the method, two benchmark examples were provided to demonstrate the accuracy and robustness of the LCG method. In general, the accuracy of the method is excellent. We believe that the LCG method has several advantages of the DGM but computing cost is lower. This is possible due to a sensible combination of a post-processing step and a time-stepping algorithm. Although, the LCG method presented here needs the formulation in space–time frame, this is not a huge disadvantage as many algorithms do use iterative methods. The LCG method produces an element-by-element solution, and further extension of the method to real practical problems with interfaces is essential to bring out the real advantages of the LCG method. It will also be interesting to extend the method to three-dimensional flow problems and to implicit time discretization.

ACKNOWLEDGEMENT

This work is partially supported by EPSRC grant E/D070554/01.

REFERENCES

1. Mizukami A. A mixed finite element method for boundary flux computation. *Computer Methods in Applied Mechanics and Engineering* 1989; **57**:239–243.
2. Gresho PM, Lee RL, Sani RL, Maslanik MK, Eaton BE. The consistent Galerkin FEM for computing derived boundary quantities in thermal and/or fluids problems. *International Journal for Numerical Methods in Fluids* 1987; **7**:371–394.
3. Hughes TJR, Franca LP, Harari I, Mallet M, Shakib R, Spelce TE. Finite element method for high-speed flows: consistent calculation of boundary flux. *Paper No. AIAA-87-0556*, *AIAA 25th Aerospace Sciences Meeting*, Reno, Nevada, 1987.
4. Oshima M, Hughes TJR, Jansen K. Consistent finite element calculation of boundary and internal fluxes. *International Journal of Computational Fluid Dynamics* 1998; **9**:227.
5. Hughes TRJ, Engel G, Mazzei L, Larson MG. The continuous Galerkin method is locally conservative. *Journal of Computational Physics* 2000; **163**:467–488.
6. Hughes TRJ, Wells GN. Conservation properties for the Galerkin and stabilised forms of the advection–diffusion and incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:1141–1159.
7. Nithiarasu P. A simple locally conservative Galerkin (LCG) finite element method for transient conservation equations. *Numerical Heat Transfer*, *Part B Fundamentals* 2004; **46**:357–370.
8. Thomas CG, Nithiarasu P. An element-wise, locally conservative Galerkin (LCG) method for diffusion and convection–diffusion problems. *International Journal for Numerical Methods in Engineering* 2006; DOI: 10.1002/nme.2095.
9. Thomas CG. A locally conservative Galerkin (LCG) method for convection–diffusion and Navier–Stokes equations. *Ph.D. Thesis*, Civil and Computational Engineering Centre, School of Engineering, Swansea University, 2006.
10. Nithiarasu P. A unified fractional step method for compressible and incompressible flows, heat transfer and incompressible solid mechanics. *International Journal of Numerical Methods for Heat and Fluid Flow* 2007; in press.
11. Massarotti N, Nithiarasu P, Zienkiewicz OC. Natural convection in porous medium-fluid, interface problems. A finite element analysis by using the CBS procedure. *International Journal of Numerical Methods for Heat and Fluid Flow* 2001; **11**:473–490.
12. Ohayon R, Felippa C. Special issue: advances in computational methods for fluid–structure interaction and coupled problems. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2977–2978.

13. Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. *Technical Report LA-UR-73-479*, 1973.
14. Lesaint P, Raviart PA. On a finite element method for solving the neutron transport equation. *Mathematical Aspects of Finite Elements in Partial Differential Equations*, *Proceedings of a Symposium Conducted by the Mathematic Research Center*, University of Wisconsin-Madison, 1–3 April 1974.
15. Chavent G, Salzano G. A finite element method for the 1d water flooding problem with gravity. *Journal of Computational Physics* 1982; **45**:307–344.
16. Chavent G, Cockburn B. The local projection $p^0 p^1$-discontinuous Galerkin finite element method for scalar conservation laws. *RAIRO—Modélisation Mathematique et Analyse Numerique* 1989; **23**:565–592.
17. Cockburn B, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws ii: General framework. *Mathematics of Computation* 1989; **52**:411–435.
18. Cockburn B, Lin SY, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws iii: one dimensional systems. *Journal of Computational Physics* 1989; **84**:90–113.
19. Cockburn B, Hou S, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws iv: the multidimensional case. *Mathematics of Computation* 1989; **54**:545–581.
20. Cockburn B, Shu CW. The Runge–Kutta local projection p^1 discontinuous Galerkin finite element method for scalar conservation laws. *RAIRO—Modélisation Mathematique et Analyse Numerique* 1991; **25**:337–361.
21. Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin finite element method for conservation laws v: multidimensional systems. *Journal of Computational Physics* 1998; **141**:337–361.
22. Cockburn B. Discontinuous Galerkin methods for convection dominated problems. *High-Order Methods for Computational Science and Engineering*. Springer: Berlin, 1999.
23. Cockburn B, Karniadakis GE, Shu C-W (eds). *The Development of Discontinuous Galerkin Methods. Discontinuous Galerkin Method. Theory*, *Computation and Applications*. Lecture Notes in Computational Science and Engineering. Springer: Berlin, 2000.
24. Cockburn B, Kanschat G, Schötzau D. The local discontinuous Galerkin methods for linear incompressible flow: a review, computer and fluids (special issue: residual based methods and discontinuous Galerkin schemes) *Computers and Fluids* 2005; **34**:491–506.
25. Zienkiewicz OC, Taylor RL, Sherwin SJ, Peiro J. On discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering* 2003; **58**:1119–1148.
26. Nithiarasu P. An efficient artificial compressibility (AC) scheme based on the characteristic based split (CBS) method for incompressible flows. *International Journal for Numerical Methods in Engineering* 2003; **56**: 1815–1845.
27. Nithiarasu P, Mathur JS, Weatherill NP, Morgan KK. Three dimensional incompressible flow calculations using the characteristic based split (CBS) scheme. *International Journal for Numerical Methods in Fluids* 2004; **44**: 1207–1229.
28. Malan AG, Lewis RW, Nithiarasu P. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: part i. Theory and implementation. *International Journal for Numerical Methods in Engineering* 2002; **54**:695–714.
29. Zienkiewicz OC, Taylor RL, Nithiarasu P. *The Finite Element Method for Fluid Dynamics*. Elsevier, Butterworths, Heinemann: London, 2005.
30. Drikakis D, Govatsos PA, Papantonis DE. A characteristic-based method for incompressible flows. *International Journal for Numerical Methods in Fluids* 1994; **19**:667–685.
31. Nithiarasu P, Codina R, Zienkiewicz OC. The characteristic based split scheme—a unified approach to fluid dynamics. *International Journal for Numerical Methods in Engineering* 2006; **66**:1514–1546.
32. Nithiarasu P, Zienkiewicz OC. Analysis of an explicit and matrix free fractional step method for incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5537–5551.
33. Zienkiewicz OC, Codina R. A general algorithm for compressible flow—part i. The split characteristic-based scheme. *International Journal for Numerical Methods in Fluids* 1995; **20**:869–885.
34. Zienkiewicz OC, Morgan K, Sai BVKS, Codina R, Vazquez M. A general algorithm for compressible flow—part ii. Tests on the explicit form. *International Journal for Numerical Methods in Fluids* 1995; **20**:887–913.
35. Nithiarasu P. A matrix free fractional step method for static and dynamic incompressible solid mechanics. *International Journal for Computational Methods in Engineering Science and Mechanics* 2006; **7**:369–380.
36. Nithiarasu P, Liu C-B, Massarotti N. Laminar and turbulent flow through a model human upper airway. *Communications in Numerical Methods in Engineering* 2007. DOI: 10.1002/cnm.939.
37. Löhner R, Morgan KK, Zienkiewicz OC. The solution of non-linear hyperbolic equation systems by the finite element method. *International Journal for Numerical Methods in Fluids* 1984; **4**:1043–1063.

38. Malan AG, Lewis RW, Nithiarasu P. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part II. Application. *International Journal for Numerical Methods in Engineering* 2002; **54**:715–729.
39. Zienkiewicz OC, Taylor RL, Zhu JZ. *The Finite Element Method. Its Basis & Fundamentals*. Elsevier, Butterworths, Heinemann: London, 2005.
40. Gaitonde AL. A dual-time method for two-dimensional unsteady incompressible flow calculations. *International Journal for Numerical Methods in Engineering* 1998; **41**:1153–1166.
41. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
42. de Sampaio PAB, Lyra PRM, Morgan K, Weatherill NP. Petrov Galerkin solutions of the incompressible Navier–Stokes equation in primitive variables with adaptive remeshing. *Computer Methods in Applied Mechanics and Engineering* 1992; **106**:143–178.
43. Panton RC. *Incompressible Flow*. Wiley: New York, 1984.
44. Schlichting H. *Boundary Layer Theory* (6th edn). McGraw-Hill: New York, 1968.
45. Codina R, Owen HC, Nithiarasu P, Liu CB. Numerical comparison of CBS and SGS as stabilization techniques for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering* 2006; **66**:1672–1689.

DOI: 10.1002/fld