

# DEVELOPMENT OF A HIGH ORDER INCOMPRESSIBLE DISCONTINUOUS GALERKIN FINITE ELEMENT SOLVER

E. Ferrer<sup>\*†</sup> and R. H. J. Willden<sup>\*††</sup>

<sup>\*</sup>Department of Engineering Science, University of Oxford,  
 Parks Road, Oxford, OX1 3PJ, UK

<sup>†</sup>e-mail: esteban.ferrer@eng.ox.ac.uk

<sup>††</sup> e-mail: richard.willden@eng.ox.ac.uk

**Key words:** Discontinuous Galerkin, Interior Penalty, splitting method, modal basis

**Abstract.** *An unsteady high order Discontinuous Galerkin (DG) code has been developed, verified and validated for the solution of the two-dimensional incompressible Navier-Stokes equations. A second order stiffly stable method has been used to discretise the equations in time. Spatial discretisation is accomplished using a modal DG approach, in which the inter-element fluxes are approximated using the Interior Penalty formulation. Three variants, Symmetric Interior Penalty Galerkin (SIPG), Incomplete Interior Penalty Galerkin (IIPG) and Nonsymmetric Interior Penalty Galerkin (NIPG), have been implemented and compared. The non-linear terms in the Navier-Stokes equations are expressed in the convective form and approximated through the Lesaint-Raviart fluxes modified for DG methods.*

*The resulting method leads to a stable scheme for the unsteady Stokes and Navier-Stokes problems when equal order approximation is used for velocity and pressure and for all fluxes tested. For the full Navier-Stokes equations, two laminar test cases are considered for the square cylinder problem at Reynolds numbers of 10 (steady wake) and 100 (unsteady wake). The results are compared to the h/p Spectral code Nektar and the commercial Finite Volume code Fluent. The developed DG code shows similar convergence trends to Nektar for the test problems considered. For the unsteady wake case, the number of degrees of freedom necessary for Fluent to reach comparable accuracy is three times larger than for the two high order methods considered.*

## 1 INTRODUCTION

The development of a numerical method requires verification and validation. Verification is performed with simplified problems where the exact solution is known and hence the error between the exact solution and the numerical solution can be quantified. Validation identifies the ability for the method to provide valuable results for real flows found in realistic engineering problems. This paper reports the verification and validation of a high order Discontinuous Galerkin (DG) method for solving the incompressible Navier-Stokes

(NS) equations in two dimensions.

The numerical solution of the incompressible NS equations requires discretisation in time (temporal derivatives) and space (spatial derivatives). Spatial and temporal derivatives can generally be decoupled leading to the *method of lines*. We select to discretise temporally using finite differences (here a dual splitting method), and a high order DG method for spatial discretisation.

In the context of spatial discretisation, high order methods ( $k > 3$ , where  $k$  is the polynomial order) are becoming of interest for problems where low spatial numerical dissipation and diffusion are required<sup>[14][15]</sup>. Through the use of high polynomial order basis functions, these methods can achieve high accuracy on a coarse mesh providing that the solution is smooth. On the other hand, low order methods ( $k \leq 3$ ) generally require a very large number of elements to reduce spatial discretisation errors to the same levels. Hence, the required overall number of degrees of freedom (DOF) for a given problem can be significantly reduced by the use of a high order method (see section 4.2).

DG methods are a family of high order methods where the continuity constraint, generally imposed in finite element and high order conformal methods (e.g. Spectral methods), is relaxed, allowing for discontinuities in the numerical solution between neighbouring elements. This flexibility is an attractive feature because  $h$ -refinement (i.e. mesh refinement) or  $p$ -refinement (i.e. polynomial order enrichment) can be performed independently and locally without the necessity of ensuring continuity across elements. Various DG type methods for elliptic type problems exist<sup>[3],[9],[16]</sup>, which differ in their treatment of the numerical fluxes used to communicate the information through discontinuous element interfaces. The Interior Penalty (IP) method is a subfamily of the DG methods that uses a penalisation technique to enforce a degree of smoothness in the numerical solution. IP methods are characterised by the necessity of tuning a penalty parameter. The solution accuracy is found to be influenced by this parameter<sup>[5]</sup> and it therefore becomes necessary to study its influence on the verification stage of the code development (see section 3).

To perform the temporal discretisation of the incompressible NS equations, we chose a fractional-step method (i.e. dual splitting technique) which relies on the Helmholtz decomposition<sup>[10]</sup>. This states that any vector field can be decomposed into an irrotational and a solenoidal component. A review of methods of this type can be found in Deville et al.<sup>[7]</sup>, Guermond et al.<sup>[10]</sup> or Karniadakis and Sherwin<sup>[14]</sup>. In particular, the code developed here uses the splitting projection scheme detailed in Karniadakis et al.<sup>[13]</sup>. This temporal scheme has been previously used in conjunction with high order continuous modal Spectral<sup>[13]</sup> and high order nodal DG<sup>[11]</sup> spatial discretisations, but to the authors knowledge, not with the DG-IP formulation in combination with modal basis functions. As in this work, the spectral continuous and nodal discontinuous formulations yielded stable solutions when the polynomial order for the velocity and pressure approximations were the same. We first describe in section 2 the methodology used which requires temporal (section 2.1) and spatial discretisation (section 2.2). The temporal discretisation leads to a decoupled system for the velocity and pressure and requires the solution of elliptic

problems that encompass the Poisson and Helmholtz equations. The solution of these equations using the DG-IP formulation is described in section 2.2.3. Section 3 is devoted to the verification of the method. We verify the method for the Poisson and Helmholtz equations, followed by the unsteady Stokes problem (without non-linear terms) where we analyse the influence of the penalty parameter on the solution accuracy. The final verification case tests the code for the full NS equations for a simple problem (Taylor Vortex), where the non-linear terms are included. Finally in Section 4, we validate the method with the square cylinder problem for two laminar cases: Reynolds number ( $Re$ ) of 10, that leads to a steady wake and  $Re = 100$  that results in unsteady periodic vortex shedding. We compare the results to the continuous Spectral code Nektar<sup>[14]</sup> and the Finite Volume code Fluent<sup>[2]</sup>.

## 2 METHODOLOGY

The two dimensional incompressible NS equations can be written using the convective formulation for the non-linear terms (in the absence of body forces):

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + \nu \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \tag{1}$$

where  $\mathbf{u} = (u, v)^T$  is the vector of the velocity components in two dimensions,  $p$  represents the static pressure and  $\nu$  the kinematic viscosity.

### 2.1 Temporal discretisation of the incompressible NS equations

A second order dual stiffly stable method<sup>[13]</sup> has been used to discretise the equations in time. Within this scheme the non-linear terms are treated explicitly whilst viscous and pressure terms are handled implicitly. The resulting temporally discretised momentum equation is given by:

$$\frac{\gamma_0 \mathbf{u}^{n+1} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = -\nabla p^{n+1} - \beta_0 [(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n] - \beta_1 [(\mathbf{u}^{n-1} \cdot \nabla) \mathbf{u}^{n-1}] + \nu \nabla^2 \mathbf{u}^{n+1}, \tag{2}$$

where  $\gamma_0$ ,  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$  and  $\beta_1$  are constants chosen to achieve the required level of temporal accuracy (see the next section) and the indices  $n-1$ ,  $n$  and  $n+1$  refer to subsequent steps separated in time by an increment  $\Delta t$ . The present method allows, by the use of intermediate variables, the incompressible NS equations to be split into three distinct equations that can be solved successively at each time-step; an explicit non-linear convection equation, a Poisson equation for pressure and a Helmholtz equation to account for the viscous terms.

#### 2.1.1 STEP 1 - Non-linear term

The first step consists of the explicit integration of the non-linear convective term using the Adams-Bashforth second order scheme. Introducing the intermediate velocity  $\tilde{\mathbf{u}}$ , the

first step can be written as:

$$\frac{\gamma_0 \tilde{\mathbf{u}} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = -\beta_0 [(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n] - \beta_1 [(\mathbf{u}^{n-1} \cdot \nabla) \mathbf{u}^{n-1}]. \quad (3)$$

By choosing  $\gamma_0 = 3/2$ ,  $\alpha_0 = 2$ ,  $\beta_0 = 2$ ,  $\alpha_1 = -1/2$  and  $\beta_1 = -1$ , the method is second order in time for the velocity. However, this second order method is not self-starting. To start the scheme one can use:  $\gamma_0 = \alpha_0 = \beta_0 = 1$  and  $\alpha_1 = \beta_1 = 0$  which reduces to the first order explicit forward Euler method.

### 2.1.2 STEP 2 - Pressure term

The second step treats the pressure term implicitly whilst imposing the divergence-free constraint on the intermediate velocity, resulting in a Poisson equation for the pressure. We introduce a secondary intermediate solenoidal velocity  $\tilde{\tilde{\mathbf{u}}}$  and write:

$$\gamma_0 \frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t} = -\nabla p^{n+1}. \quad (4)$$

Taking the divergence of Eq. 4, and imposing the divergence-free constraint on  $\tilde{\tilde{\mathbf{u}}}$  (i.e.  $\nabla \cdot \tilde{\tilde{\mathbf{u}}} = 0$ ) we obtain:

$$-\gamma_0 \frac{1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}} = -\nabla^2 p^{n+1}. \quad (5)$$

This is a Poisson problem that can be closed using a Neumann boundary condition at inflow and wall boundaries derived from taking the dot product of the outward normal vector at the boundaries  $\mathbf{n}$  with the momentum equation;

$$\begin{aligned} \frac{\partial p^{n+1}}{\partial n} = & -\beta_0 \mathbf{n} \cdot \left( \frac{\partial \mathbf{u}^n}{\partial t} + (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nu \nabla \times \boldsymbol{\omega}^n \right) \\ & - \beta_1 \mathbf{n} \cdot \left( \frac{\partial \mathbf{u}^{n-1}}{\partial t} + (\mathbf{u}^{n-1} \cdot \nabla) \mathbf{u}^{n-1} + \nu \nabla \times \boldsymbol{\omega}^{n-1} \right), \end{aligned} \quad (6)$$

where we have used the vorticity definition ( $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ ) and the identity  $\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u}$  which for an incompressible flow reduces to  $\nabla^2 \mathbf{u} = -\nabla \times \nabla \times \mathbf{u} = -\nabla \times \boldsymbol{\omega}$ . This expression indirectly enforces the incompressibility constraint and gives a consistent method<sup>[14]</sup>. We note that this definition of the boundary condition requires that the velocity is twice differentiable, and is one of reasons why this method is generally used with high order methods and not with low order finite elements<sup>[10]</sup>. Once the Poisson equation is solved, we can use  $p^{n+1}$  to update the intermediate velocity field by rearranging Eq. 4.

### 2.1.3 STEP 3 - Viscous Term

Finally, this step accounts for the implicit treatment of the viscous term, leading to a Helmholtz equation for the velocity. The final step reads:

$$\gamma_0 \left( \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\Delta t} \right) = \nu \nabla^2 \mathbf{u}^{n+1}, \quad (7)$$

which can be written as a Helmholtz equation:

$$-\nabla^2 \mathbf{u}^{n+1} + \frac{\gamma_0}{\nu \Delta t} \mathbf{u}^{n+1} = \frac{\gamma_0 \tilde{\mathbf{u}}}{\nu \Delta t}. \quad (8)$$

We note that the final velocity is not enforced to be explicitly divergence free, since the explicit imposition of the divergence free constraint has been imposed in step 2 on  $\tilde{\mathbf{u}}$ .

To summarise, the temporal discretisation requires the solution of the three steps defined to advance the solution in time. It can be easily seen that when adding Eq. 3, Eq. 4 and Eq. 7 we recover the original momentum equation discretised in time Eq. 2.

## 2.2 Spatial discretisation

The previous temporal scheme requires the spatial solution of non-linear terms (step 1) and of elliptic equations (steps 2 and 3) at each time step. This section summarises the DG spatial discretisation of these terms.

### 2.2.1 Preliminaries: Spaces and traces

Let  $\Omega$  be a polygonal domain in  $\mathbb{R}^2$  with boundaries  $\partial\Omega$  that can be of Dirichlet type ( $\Gamma_D$ ) or Neumann type ( $\Gamma_N$ ). We introduce a triangular tessellation  $\Omega_h = \{el\}$  of  $\Omega$  with external boundaries  $\partial\Omega_h$  and  $\Gamma_h$  interior boundaries of  $\Omega_h$ . We further define for each  $el \in \Omega_h$ ,  $\Gamma$  as the positively oriented, i.e. anticlockwise, single edge of the element boundaries  $\partial el$  and the normal vector  $\mathbf{n}_\Gamma$  as the unit outward pointing normal.

We consider the discontinuous approximate space  $D_k(\Omega_h)$  as:

$$D_k(\Omega_h) = \{v_h \in L^2(\Omega) : \forall el \in \Omega_h, v_h|_{el} \in \mathbb{P}_k(\Omega)\}, \quad (9)$$

and its vector version  $D_k^2(\Omega_h)$  where  $\mathbb{P}_k(\Omega)$  denotes the space of polynomials of order less than or equal to  $k$ .

We also define the jump  $[[\bullet]]$  and average  $\{\{\bullet\}\}$  across the interface between neighboring elements  $el_1$  and  $el_2$ , arising from the discontinuous discretisation:

$$[[\bullet]] = (\bullet|_{el_1}) - (\bullet|_{el_2}), \quad \{\{\bullet\}\} = \frac{1}{2}(\bullet|_{el_1}) + \frac{1}{2}(\bullet|_{el_2}), \quad \forall \Gamma = \partial el_1 \cap \partial el_2, \quad (10)$$

and at boundary edges as:

$$[[\bullet]] = \{\{\bullet\}\} = (\bullet|_{el_1}), \quad \forall \Gamma = \partial el_1 \cap \partial\Omega, \quad (11)$$

### 2.2.2 Spatial discretisation of the Non-linear terms

The non-linear term in the Navier-Stokes equations is expressed in the convective form and approximated through the Lesaint-Raviart fluxes modified for DG methods<sup>[9],[16]</sup>, where upwinding is performed to account for convection dominated flows. If we consider the discretised version of the non-linear terms integrated over  $\Omega_h$ , we find  $\mathbf{u}_h \in D_k^2(\Omega_h)$  such that:

$$\begin{aligned} \int_{\Omega_h} \mathbf{u}_h \cdot \nabla \mathbf{u}_h \cdot \mathbf{v}_h d\mathbf{x} &= \sum_{el \in \Omega_h} \int_{el} (\mathbf{u}_h \cdot \nabla \mathbf{u}_h \cdot \mathbf{v}_h) d\mathbf{x} + 1/2 \sum_{el \in \Omega_h} \int_{el} (\nabla \cdot \mathbf{u}_h) \mathbf{u}_h \cdot \mathbf{v}_h d\mathbf{x} \\ &+ \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_{\Gamma^-} |\{\{\mathbf{u}_h\}\} \cdot \mathbf{n}_\Gamma| (\mathbf{u}_h^{int} - \mathbf{u}_h^{ext}) \cdot \mathbf{v}_h^{int} ds \\ &- 1/2 \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_{\Gamma} [[\mathbf{u}_h]] \cdot \mathbf{n}_\Gamma \{\{\mathbf{u}_h \cdot \mathbf{v}_h\}\} ds, \quad \forall \mathbf{v}_h \in D_k^2(\Omega_h). \end{aligned} \quad (12)$$

The third term on the right hand side is the upwinding term, required for convection dominated flows. The quantity  $\mathbf{u}_h^{int} - \mathbf{u}_h^{ext}$  is another way to write the jump as done by Riviere<sup>[16]</sup>.  $\Gamma^-$  defines the portion of the boundary where the flow enters the element (i.e.  $\{\{\mathbf{u}_h\}\} \cdot \mathbf{n}_{el} < 0$ ). Since the treatment of the non-linear terms is explicit, the calculation is performed in the physical space, at quadrature points as opposed to in modal space (see section 2.2.5). The inclusion of the explicit treatment for the non-linear terms introduces a Courant-Friedrichs-Lewy (CFL) type restriction on the time step. The CFL estimate for high order spatial methods is  $h/Uk^2 > \Delta t$  as shown in Karniadakis and Sherwin<sup>[14]</sup> for an advection model problem, where  $h$  is the mesh element size,  $U$  the characteristic velocity and  $k$  represents the polynomial order.

### 2.2.3 The DG-IP spatial discretisation for elliptic equations

The described temporal splitting method requires the solution of purely elliptic equations (Poisson and Helmholtz). We choose various forms of the IP method to spatially discretise these equations: namely the Symmetric Interior Penalty Galerkin (SIPG), Incomplete Interior Penalty Galerkin (IIPG) and Nonsymmetric Interior Penalty Galerkin (NIPG) methods. The details of the development, derivation and implementation are omitted here but can be found in various references<sup>[9],[16]</sup>. All three methods can be conveniently recast in a single form which is summarised hereafter. We introduce here the continuous elliptic problem and its discretisation using DG-IP methods:

$$\begin{aligned} -\Delta u + \alpha u &= g \quad \text{in } \Omega, \\ u &= L_D \quad \text{on } \Gamma_D, \\ \nabla u \cdot \mathbf{n} &= L_N \quad \text{on } \Gamma_N, \end{aligned} \quad (13)$$

where  $u \in C^2(\Omega)$  is a scalar solution (but extension to the vector formulation is direct),  $\alpha$  is real and represents the wave number for the Helmholtz equations,  $\mathbf{n}$  is the outward

pointing unit normal vector on  $\partial\Omega$ ,  $g \in L_2(\Omega)$  is the forcing term, and  $L_D \in H^{1/2}(\Gamma_D)$  and  $L_N \in L_2(\Gamma_N)$  represent the boundary conditions. This form defines the Poisson and Helmholtz equation as follows:

- **Poisson:** forcing term  $g \neq 0$  and convective term  $\alpha = 0$
- **Helmholtz:** forcing term  $g \neq 0$  and convective term  $\alpha \neq 0$

We note that the Laplace equation (i.e.  $g = 0; \alpha = 0$ ) can also be considered with this formulation but is omitted in this text.

To find a weak solution of the continuous problem Eq. 13 using the DG-IP method, we introduce the bilinear  $a_\varepsilon(\bullet, \bullet)$  and linear  $\ell(\bullet)$  forms associated with the discrete variational formulation arising from the weak form of the continuous problem<sup>[9],[16]</sup>.

$$\begin{aligned} a_\varepsilon(v_h, u_h) = & \sum_{el \in \Omega_h} \int_{el} (\nabla v_h \cdot \nabla u_h) d\mathbf{x} + \sum_{el \in \Omega_h} \int_{el} (\alpha v_h u_h) d\mathbf{x} - \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_{\Gamma} [[v_h]] \{ \{ \nabla u_h \} \} \cdot \mathbf{n}_{\Gamma} ds \\ & + \varepsilon \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \int_{\Gamma} [[u_h]] \{ \{ \nabla v_h \} \} \cdot \mathbf{n}_{\Gamma} ds + \sum_{\Gamma \in \Gamma_h \cup \Gamma_D} \frac{\sigma}{|trace|^\beta} \int_{\Gamma} [[v_h]] [[u_h]] ds, \end{aligned} \quad (14)$$

and the following linear form:

$$\ell(v_h) = \sum_{el \in \Omega_h} \int_{el} (v_h g_h) d\mathbf{x} + \sum_{\Gamma \in \Gamma_D} \int_{\Gamma} (\varepsilon \nabla v_h \cdot \mathbf{n}_{\Gamma} + \frac{\sigma}{|trace|^\beta} v_h) L_D ds + \sum_{\Gamma \in \Gamma_N} \int_{\Gamma} v_h L_N ds, \quad (15)$$

for all  $v_h \in D_k(\Omega_h)$ , where  $\sigma$  and  $\beta$  are positive real parameters and the  $|trace|$  is defined as the length of the edge in 2D. Then, the DG-IP approximate solution reduces to finding  $u_h \in D_k(\Omega_h)$  provided by  $a_\varepsilon(v_h, u_h) = \ell(v_h)$ ,  $\forall v_h \in D_k(\Omega_h)$ .

The choice of  $\varepsilon$  in Eq. 14 and Eq. 15 defines the method and has consequences on the choice of the penalty parameters which is responsible of enforcing coercivity of the bilinear form<sup>[16]</sup>. In the following table 1 we summarise the penalty parameter requirement  $(\sigma, \beta)$  and associated a priori theoretically derived convergence rates in the  $L_2$  norm when  $h$ -refinement is performed with fixed polynomial order  $k$ <sup>[16]</sup>. We note that the penalty  $\beta$  is local in the sense that it is related to the local edge length ( $trace$ ) and that the parameter  $\sigma$  is of global nature. The relative importance of each penalty is of little importance when regular meshes are used as in the verification section.

If desired, the explicit expression for the penalty parameter derived for the SIPG method by Shahbazi<sup>[17]</sup>, may be substituted in 14 and 15:

$$\frac{\sigma}{|trace|^\beta} = \frac{(k+1)(k+d)}{d} \left[ \max \left( \frac{S_{el1}}{V_{el1}}, \frac{S_{el2}}{V_{el2}} \right) \right], \quad (16)$$

where  $k$  is the polynomial order,  $d = 2$  corresponds to the dimension of the problem,  $S_{el}$  and  $V_{el}$  are the perimeter and area (in two dimensions) of each of the elements 1 and 2 sharing the edge. This expression does not require any arbitrary penalisation term ( $\sigma, \beta$  combination), but automatically adjusts the penalisation term according to the underlying

IP Method	$\varepsilon$	$\sigma$	$L_2$ Convergence $\beta = 1$	$L_2$ Convergence $\beta = 3$
SIPG	-1	large	$O(h^{k+1})$	$O(h^{k+1})$
SIPG shahbazi	-1	-	$O(h^{k+1})$	$O(h^{k+1})$
IIPG	0	large	$O(h^{k+1})$ , $k$ odd $O(h^k)$ , $k$ even	$O(h^{k+1})$
NIPG 0	+1	= 0	$O(h^{k+1})$ , $k$ odd $O(h^k)$ , $k$ even	—
NIPG 1	+1	= 1	$O(h^{k+1})$ , $k$ odd $O(h^k)$ , $k$ even	$O(h^{k+1})$

Table 1: Penalty parameter requirements and theoretical convergence rates for various IP methods.

mesh geometry.

The table shows that the SIPG method provides optimal convergence, i.e.  $O(h^{k+1})$ , in the  $L_2$  norm independently of the penalisation used as long as this is large enough. The IIPG and NIPG methods provide optimal convergence only for odd polynomials if no super-penalisation is used, i.e.  $\beta = 1$ , but for all polynomials when using super-penalisation, i.e.  $\beta = 3$ . We note that the bilinear form associated with the SIPG method is symmetric leading to positive definite matrix but that the other fluxes lead to non symmetric systems which has consequences in the choice of the iterative solver that can be used.

#### 2.2.4 Mapping from physical space to computational space

Up to this point, the discretisation has been defined in the physical space and general vector notation has been used. This section deals with the mapping between the physical space which will be defined in cartesian coordinates  $(x, y)$  and the computational space  $(\xi_1, \xi_2)$ , where the described method is actually computed. The transformation and details for the *collapsed* coordinate system<sup>[14]</sup> are summarised in Fig. 1. The transformation is constituted by two steps. The first step (step 1) maps the the standard square  $(\xi_1, \xi_2)$  to the standard triangle  $(\eta_1, \eta_2)$  where one vertex in the square is collapsed. The second (step 2), transforms the standard triangle  $(\eta_1, \eta_2)$  to the general straight sided triangle  $(x, y)$ .

#### 2.2.5 Modal basis functions

Within each element  $el$ , the approximate solution,  $u_h$ , of the defined scheme is expressed as a modal expansion of hierarchical basis functions  $\phi_i(x, y)$  as:

$$u_h(x, y) = \sum_{i=1}^{TotModes} u_i \phi_i(x, y), \quad (17)$$



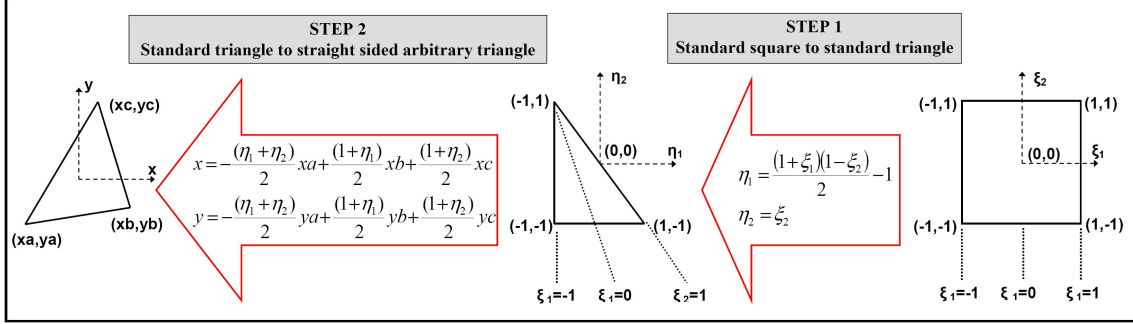


Figure 1: Mapping from computational space to physical space.

where  $x, y \in el$  and  $TotModes$  is the number of modes in each element (i.e. local number of degrees of freedom). The necessary number of modes to completely define the expansion on triangular elements is  $TotModes = (k+1)(k+2)/2$ <sup>[14]</sup>. We select a tensorial expansion basis<sup>[14]</sup> defined in the standard triangle  $(\eta_1, \eta_2)$  as follow:

$$\phi_i(x, y) = \phi_{pq}(x, y) = \phi_{pq}(\eta_1, \eta_2) = \varphi_p^a(\xi_1) \varphi_{pq}^b(\xi_2), \quad (18)$$

where  $p, q$  denote the different components of the tensorial expansion with  $0 \leq p$  and  $p+q \leq k$ , with  $k$  the polynomial order, i.e  $\mathbb{P}_k(\Omega)$ . The bases  $\varphi_p^a$  and  $\varphi_{pq}^b$  are called principal functions<sup>[14]</sup> and allow us to express the basis function  $\phi_{pq}$  defined in the standard triangle  $(\eta_1, \eta_2)$  in terms of basis functions defined in the standard square  $(\xi_1, \xi_2)$ . In this work, the principal basis functions are defined in terms of Jacobi polynomials, where  $P_p^{(\alpha, \beta)}(z)$  is the  $p$ th order Jacobi polynomial (and  $\alpha = \beta = 0$  corresponds to the Legendre polynomials). The bases have been normalised so the resulting mass matrix is the identity matrix:

$$\begin{aligned} \varphi_p^a(\xi_1) &= \sqrt{\frac{2p+1}{2}} P_p^{(0,0)}(\xi_1), \\ \varphi_{pq}^b(\xi_2) &= \sqrt{p+q+1} \left( \frac{1-\xi_2}{2} \right)^p P_q^{(2p+1,0)}(\xi_2). \end{aligned} \quad (19)$$

We note that this family of bases in combination with the described mapping, allow for the use of the efficient sum-factorisation technique<sup>[14]</sup>. These expansions do not ensure continuity across element boundaries but this is not necessary within the DG context as it is for continuous Spectral methods.

## 2.2.6 Numerical evaluation of the integrals: Quadrature

In this work we use Gaussian quadrature for both surface and line integrals. An important detail in the DG-IP method is that it is necessary to evaluate the basis at the edges of the elements when performing line integrals. This evaluation is performed in a preprocessing step. All required matrices are also computed prior to any time stepping. Care has been taken to ensure that sufficient quadrature points have been used for exact integration of each term, including non-linear terms.

### 2.3 Implementation and solution of the linear system

In all of the verification problems addressed below, direct solvers (e.g LU decomposition or Cholesky factorisation for positive definite systems) have been used to avoid inaccuracies introduced in the solution through the use of iterative solvers. In the last section, relating to the square cylinder, the MKL-Pardiso solver<sup>[12]</sup> has been used to solve the linear system. The method described has been implemented using the object oriented language C++. The grid generator used to construct the computational meshes is Gambit<sup>[2]</sup> using the Neutral format and post-processing is performed using Tecplot.

## 3 VERIFICATION OF THE METHOD

We use the verification problems to investigate the method's convergence properties and the influence of the penalty parameter in the solution accuracy. We will first consider purely elliptic problems, followed by the Stokes problem and then the full NS equations.

### 3.1 Elliptic equations

#### 3.1.1 Poisson equation

We select the Poisson problem, i.e.  $-\Delta u = g$ , solved in  $\Omega = [-1, 1]^2$ . The problem is defined with the forcing term  $g(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$  and with Dirichlet boundary conditions  $L_D = 0$ . The analytical solution is  $u^{exact}(x, y) = \sin(\pi x) \sin(\pi y)$ . We do not use super-penalisation and set the penalty parameter  $\beta = 1$  for all IP formulation. For the IIPG and SIPG methods we use  $\sigma = k(3k + 3)$  <sup>[16]</sup>. We show the exponential convergence obtained using  $p$  refinement on a grid with 200 triangular elements in Fig. 2.a. We also tabulate the convergence rates for  $h$ -refinement using four meshes constituted of 50, 200, 450 and 800 triangular elements in table 2. The slopes agree well with the a priori error estimates (see table 1). However, we note that the convergence rates for the IIPG method are better than predicted by the theory for even  $k$ .

We explore the influence of the penalty parameter,  $\sigma$ , on the solution accuracy for this particular problem and all fluxes. We fix the polynomial order to  $k = 2$ , since the effect of the penalty parameter is more apparent for low polynomial orders, and we increase the penalty parameter in Fig. 2.b. We observe that, for this particular test case, the IIPG and NIPG methods show decay of the error as  $\sigma$  is increased, but for the SIPG method

IP Method	k=2	k=3	k=4	k=5	k=6
IIPG	2.505	4.038	4.807	5.966	6.898
NIPG 0	1.853	3.847	4.021	5.914	6.276
NIPG 1	1.880	3.899	4.019	5.923	6.283
SIPG	3.054	4.016	4.957	5.958	6.966
SIPG Shahbazi	2.993	4.043	4.942	5.976	6.960

Table 2: Convergence rates for the Poisson equation when performing  $h$ -refinement

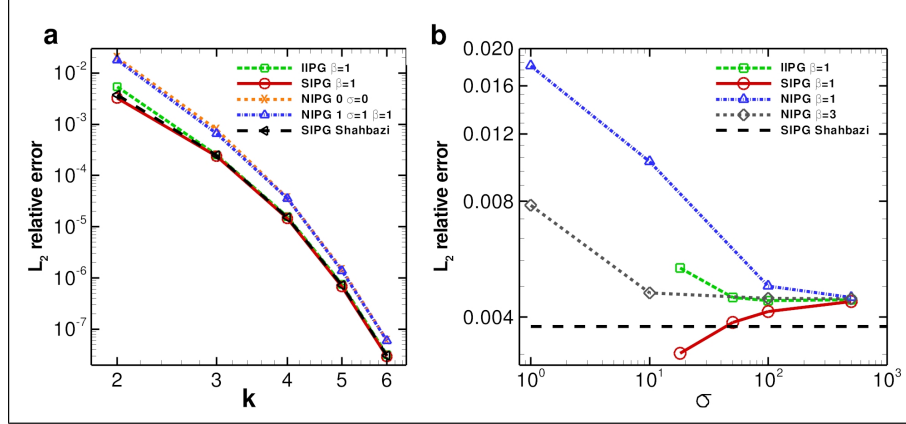


Figure 2: (a)  $L_2$  relative error norm for the Poisson equation and  $p$ -refinement; (b) Penalty parameter ( $\sigma$ ) influence on the  $L_2$  error norm for the Poisson equation,  $k=2$ .

the error increases. It is also observed that as the penalty increases all the methods converge to the same result. This fact was noted by Castillo<sup>[5]</sup> when comparing the NIPG and the SIPG methods. Finally, we note that the Shahbazi expression provides enough penalisation when compared to the SIPG method.

### 3.1.2 Helmholtz equation

The inhomogeneous Helmholtz equation, i.e.  $-\Delta u + \alpha u = g$ , is solved in  $\Omega = [0, 1]^2$ . The wave number  $\alpha$  can be chosen arbitrarily provided that the forcing term is of the form  $g(x, y) = (\alpha + 2\pi^2)\cos(\pi x)\cos(\pi y)$ . The problem is closed using Dirichlet boundary conditions derived from the exact solution  $u^{exact}(x, y) = \cos(\pi x)\cos(\pi y)$ . We fix the wave number  $\alpha = 10$ , the penalty parameter  $\beta = 1$  for all IP formulation and  $\sigma = k(3k + 3)$  for the IIPG and SIPG, and as previously, we tabulate the convergence rates for  $h$ -refinement using three meshes with 50, 200 and 338 elements in table 3. As before we see that the IIPG method provides better results than predicted by the theory showing optimal convergence for all polynomial orders. The meshes used, which are similar those used to solve the previous Poisson problem is of the type shown in Fig. 3.a.

We again investigate the effect of the penalty parameter on the  $L_2$  error accuracy for the SIPG method and a fixed polynomial order  $k = 2$  and  $\beta = 1$ . This time we vary the

IP Method	k=2	k=3	k=4	k=5	k=6
IIPG	2.858	4.035	4.937	5.989	6.953
NIPG 0	2.452	3.925	4.275	5.994	6.469
NIPG 1	2.211	3.960	4.274	5.996	6.475
SIPG	3.000	4.019	4.981	5.978	6.984
SIPG Shahbazi	2.957	4.038	4.975	5.993	6.982

Table 3: Convergence rates for the Helmholtz equation with  $\alpha = 10$  when performing  $h$ -refinement

wave number  $0 \leq \alpha \leq 10^5$  and the penalty  $18 \leq \sigma \leq 10^4$  as shown in Fig. 3.b. It is interesting to analyse the behaviour of the method for large wave numbers because when considering the Helmholtz step in the splitting scheme, the wave number corresponds to  $\alpha = \gamma_0/(\nu \Delta t)$ , which can become large for small time steps. As for the Poisson problem,

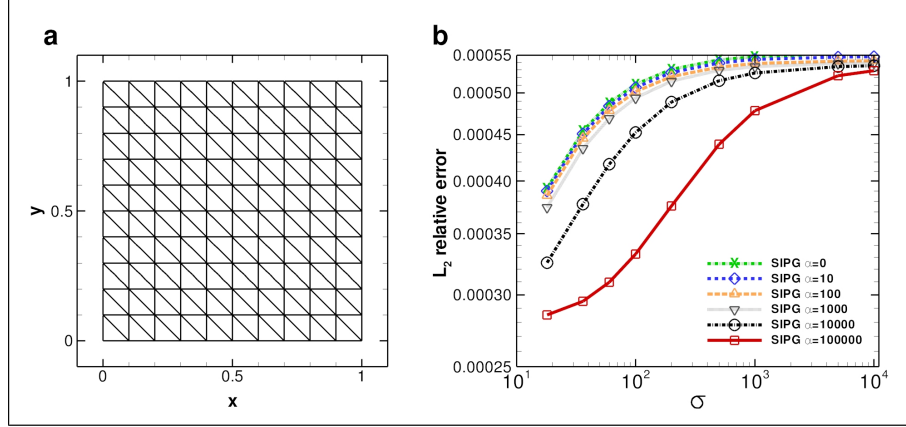


Figure 3: (a) 200 element mesh used to study the influence of the penalty parameter  $\sigma$  on the solution of the Helmholtz equation; (b) Penalty parameter ( $\sigma$ ) influence on the  $L_2$  error norm for the Helmholtz equation of various wave numbers ( $\alpha$ ), SIPG flux and  $k = 2$ .

we observe an increase of the  $L_2$  error norm as  $\sigma$  increases. It can also be seen that for large wave numbers, the penalty has greater effect.

### 3.2 Unsteady Stokes equations

We note that the time discretisation splitting scheme defined in section 2.1 can be used when the non-linear term is not considered. In this case, the continuous momentum equation reduces to  $\partial \mathbf{u} / \partial t = -\nabla p + \nu \nabla^2 \mathbf{u}$ . We consider the following solution of the Stokes problem<sup>[18]</sup>:

$$\begin{aligned} (u, v) &= (\sin(x)(a \sin(ay) - \cos(a) \sinh(y)), \cos(x)(\cos(ay) - \cos(a) \cosh(y))) e^{-\lambda t}, \\ p &= \lambda \cos(a) \cos(x) \sinh(y) e^{-\lambda t}, \end{aligned} \quad (20)$$

with  $a = 2.883356$  and  $\lambda = 9.313739$ . This unsteady problem can be solved in  $\Omega = [-1, 1]^2$  with Dirichlet boundary conditions and the initial condition provided by the exact solution. The tests have been carried out using a mesh constituted of 72 triangular elements (similar to the mesh depicted in Fig. 3.a) and the penalties  $\beta = 1$  and  $\sigma = k(3k + 3)$  when not stated otherwise. We only consider equal order polynomial spaces  $\mathbb{P}_k(\Omega)$ - $\mathbb{P}_k(\Omega)$  for velocity and pressure. The following Fig. 4.a and Fig. 4.b show the  $L_2$  convergence obtained for time step refinement for the SIPG method for the pressure and the velocity respectively. For spatially well resolved simulations ( $k \geq 7$ ), we obtain a convergence rate for the  $L_2$  relative error norm for pressure of 1.82 and 2.21 for velocity. These convergence rates are the same for all fluxes as long as the spatial discretisation is

fine enough (reduced spatial error using either  $p$  or  $h$  refinement). The figures show that for all polynomial orders the scheme provides convergent solutions as long as the time step remains larger than a certain  $\Delta t_{min}$  which reduces as  $k$  increases. The overall error seems to behave as  $\mathcal{O}(\Delta t^2 + h^{k+1}/\Delta t)$  with  $h$  representing the element size, similarly to the semi-Lagrangian method reported in Karniadakis and Sherwin<sup>[14]</sup>.

In Fig. 4.c and Fig. 4.d, we depict the solution for  $k = 2$  and 3 and two different values for the penalty  $\sigma$ . We observe an increase on the  $L_2$  norm (negative effect) for moderate time step reduction as expected from the effect of the penalisation in previous elliptic problems. However, we also see a positive effect (i.e. smaller  $L_2$  error norm) for low time

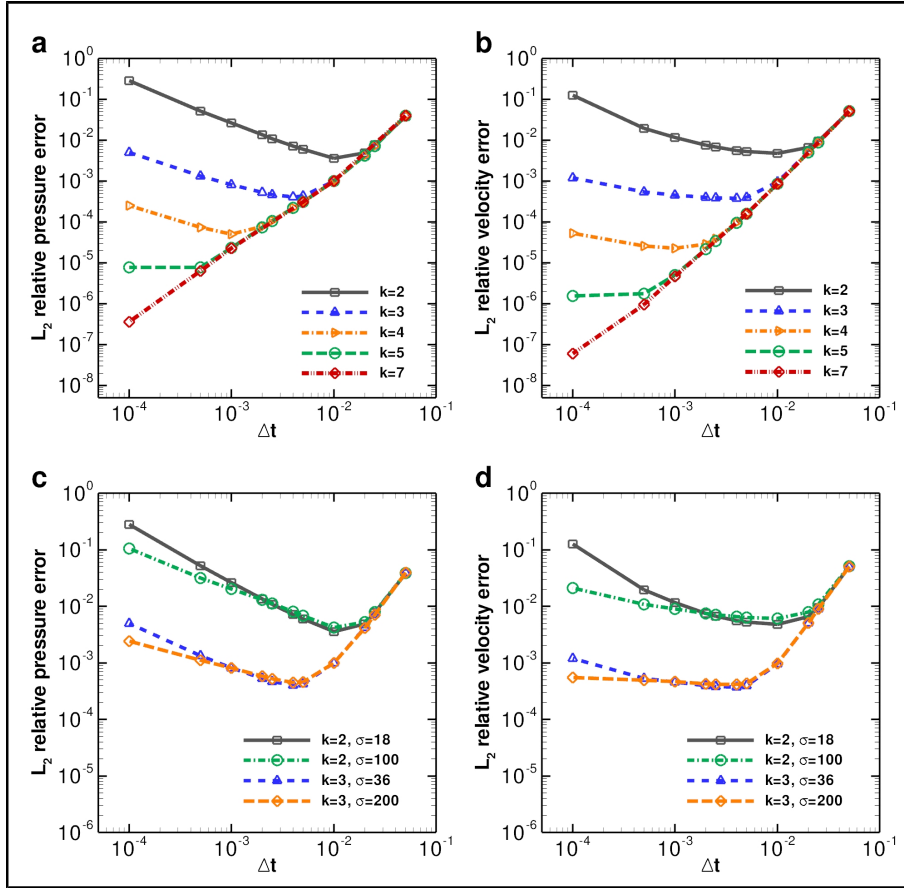


Figure 4: Time step refinement study for the Stokes equations at final time=0.1, for various polynomial orders. (a) SIPG flux:  $L_2$  relative error norm for pressure, (b) SIPG flux:  $L_2$  relative error norm for velocity, (c) SIPG flux:  $L_2$  relative error norm for pressure with various penalisation ( $\sigma$ ), (d) SIPG flux:  $L_2$  relative error norm for pressure with various penalisation ( $\sigma$ ).

steps as we increase  $\sigma$ . This last  $\Delta t$  region is however divergent in the sense that the error becomes unbounded with time. In what follows we ensure that spatial discretisation is fine enough for the selected time step ( $\Delta t > \Delta t_{min}$ ) thus avoiding the  $L_2$  error increase described here.

### 3.3 NS equations: Taylor Vortex

The selected problem to test the full incompressible NS equations including the non-linear terms is the so called Taylor vortex<sup>[18],[14]</sup>. The time dependent solution that is used to impose Dirichlet boundary conditions and the initial condition is given by:

$$\begin{aligned}(u, v) &= (-\cos(\pi x)\sin(\pi y), \sin(\pi x)\cos(\pi y)) e^{-2\nu\pi^2 t}, \\ p &= -\frac{1}{4}(\cos(2\pi x) + \cos(2\pi y)) e^{-4\nu\pi^2 t}.\end{aligned}\tag{21}$$

The problem is solved in  $\Omega = [-1, 1]^2$  discretised with 72 triangular elements with the kinematic viscosity  $\nu = 0.02$ . We again select the SIPG method ( $\beta = 1$  and  $\sigma = k(3k+3)$ ) and equal order elements for velocity and pressure. We perform  $p$ -refinement and plot the  $L_2$  (Fig. 5.a) and maximum pointwise error (Fig. 5.b) showing exponential convergence.

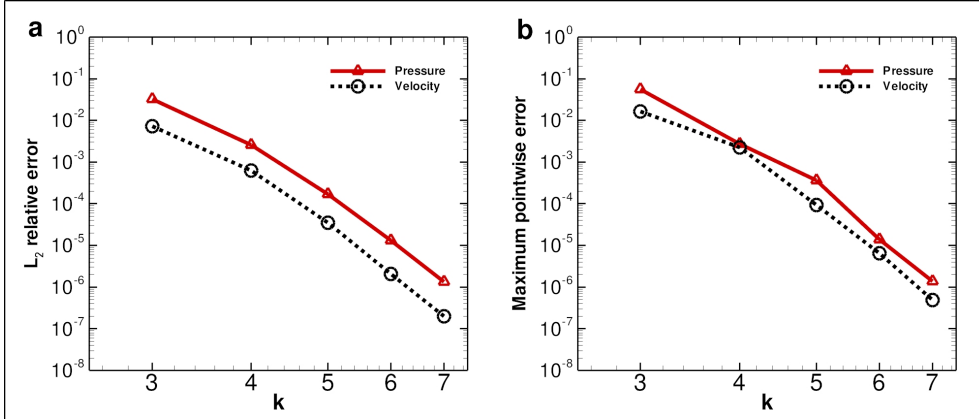


Figure 5: (a)  $L_2$  relative error norm and (b) maximum pointwise error convergence rates for  $p$ -refinement and the Taylor Vortex problem,  $\Delta t = 0.001$  and final time=0.1.

## 4 VALIDATION OF THE METHOD

The square cylinder has been selected as a test case to validate the solver in a complex flow. The selected Reynolds numbers are  $Re = 10$  and  $Re = 100$  which are low enough for the flow to remain laminar. For low Reynolds numbers ( $Re_{crit} \lesssim 50$ )<sup>[4]</sup>, there is no vortex shedding and thus the solution is steady in time, enabling validation of the code in steady flow conditions. For the higher Reynolds number, vortex shedding develops forming a Von Kármán vortex street with a characteristic vortex shedding frequency; the Strouhal number ( $St$ ). The square cylinder case has been previously studied by Darekar and Sherwin<sup>[6]</sup> using the Spectral code Nektar<sup>[14]</sup> and by Shahbazi et al.<sup>[18]</sup> where they compared their DG method to Darekar and Sherwin's results. The domain boundaries and effective blockage ratio ( $B = 2.3\%$ ) are identical to those used in the previous referenced studies. We use equal order  $P_k(\Omega)$ - $P_k(\Omega)$  elements for velocity and pressure and a mesh constituted of 836 triangular elements. We concentrate on the SIPG flux with  $\beta = 1$

although the other fluxes have shown very similar results. Since in this case there is a very significant variation of the element edge lengths ( $\sim 0.01 - 10$ ), we require an increase in the penalty parameter  $\sigma$  to ensure that the term  $\frac{\sigma}{|trace|^\beta}$  is large enough everywhere. A typical flowfield for  $Re = 100$  is shown in Fig. 6.

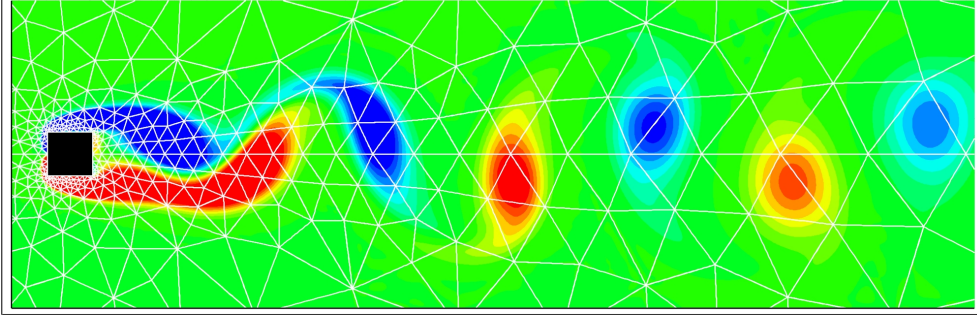


Figure 6: DG solution: Vorticity contours  $-1 \leq \omega_z \leq 1$  for the square cylinder at  $Re=100$ ,  $k = 7$ , SIPG flux. The unstructured triangular mesh is overlaid in the figure.

#### 4.1 Post processing forces- LIFT-DRAG

Lift and drag forces can be calculated using various formulations. In particular compressible or incompressible formulations (i.e. the velocity is assumed divergence free) can lead to different results. The temporal discretisation used within this work leads to a solution for the velocity that is not explicitly solenoidal at the end of the time step advancement, since the divergence free constraint is imposed on an intermediary velocity field (i.e.  $\nabla \cdot \tilde{\mathbf{u}} = 0$ ). We consider three formulations for calculating lift and drag forces. The pressure component of the forces is the same for all formulations and only the treatment of the viscous terms differs.

- **Compressible formulation**<sup>[1]</sup>:

$$\begin{aligned} F_x^C &= \oint -pn_x ds + \oint \left[ \lambda(\nabla \cdot \mathbf{u}) + 2\mu \frac{\partial u}{\partial x} \right] n_x ds + \oint \mu \left[ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] n_y ds, \\ F_y^C &= \oint -pn_y ds + \oint \left[ \lambda(\nabla \cdot \mathbf{u}) + 2\mu \frac{\partial v}{\partial y} \right] n_y ds + \oint \mu \left[ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] n_x ds, \end{aligned} \quad (22)$$

where  $\lambda = -2/3\mu$  is the second viscosity coefficient and the  $s$  parametrises the closed contour around the body.

- **Incompressible formulation:** This definition, assumes that the velocity is divergence free and sets the term  $\nabla \cdot \mathbf{u} = 0$ :

$$\begin{aligned} F_x^I &= \oint -pn_x ds + \oint \left[ 2\mu \frac{\partial u}{\partial x} \right] n_x ds + \oint \mu \left[ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] n_y ds, \\ F_y^I &= \oint -pn_y ds + \oint \left[ 2\mu \frac{\partial v}{\partial y} \right] n_y ds + \oint \mu \left[ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] n_x ds, \end{aligned} \quad (23)$$



This definition has been used by Hesthaven and Warburton<sup>[11]</sup> and is implemented in the Spectral code Nektar<sup>[14]</sup> to compute lift and drag forces.

- **Vorticity formulation:** We can derive an expression for the lift and drag viscous components using vorticity, which for 2D flows reduces to  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z) = \left(0, 0, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)$  in 2D flows and reads:

$$\begin{aligned} F_x^V &= \oint -pn_x ds + \oint \mu \omega_z n_y ds, \\ F_y^V &= \oint -pn_y ds - \oint \mu \omega_z n_x ds, \end{aligned} \quad (24)$$

To assess the influence of the formulation, we consider the steady state case of flow past the square section cylinder at  $Re = 10$  and compare the viscous forces using the three formulations while performing  $p$  refinement. Note that this is a comparison of post-processing techniques and the underlying flow-field is identical for each post-processing formulation. We also compare the results obtained using the DG formulation to the high order Spectral code Nektar. We note that the second order temporal splitting scheme used by Nektar is the same as that used for the present DG implementation but that the space discretisation is different (i.e. continuous vs discontinuous).

We show in Fig. 7.a and 7.b the viscous lift and viscous drag components respectively. The viscous lift for this steady case should be identically zero, which follows from the

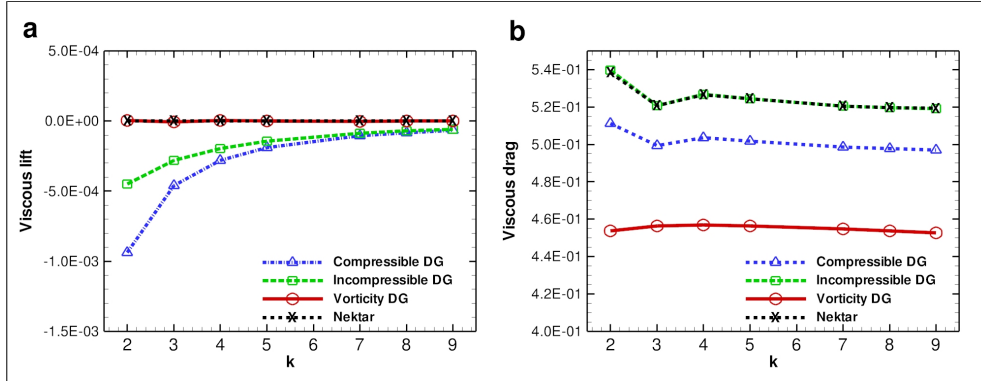


Figure 7: Square section cylinder at  $Re=10$ : (a) Viscous lift force and (b) viscous drag force as a function of the polynomial order  $k$  for the compressible, incompressible and vorticity formulations of the DG code with SIPG flux, and the continuous Spectral code Nektar.

symmetry of the mesh used. We note that the continuous Spectral code Nektar gives exactly zero lift independently of the polynomial order. All of the tested formulations provide a lift force which is smaller than  $10^{-3}$  for  $k = 2$  and tend to zero as we perform  $p$ -refinement. The formulation that seems to provide the smallest viscous lift is the vorticity formulation for all  $k$ . Indeed, if the quadrature points used to calculate the line integrals for the lift force are increased, the compressible and incompressible formulations give



a lift force that is closer to zero. However the vorticity formulation appears to be less sensitive to the number of quadrature points and provides an accurate lift force when using fewer quadrature points. When comparing the viscous drag in Fig. 7.b, we observe that DG and Nektar results agree for the incompressible formulation. When comparing the different post-processing formulations, we observe a consistent difference for all polynomial orders tested (e.g. for  $k=4$ , the difference between the compressible formulation and the incompressible is -4.40% and -13.24% for the vorticity formulation). This difference is not influenced by the number of quadrature points but reduces if the time step is reduced (not shown). The vorticity formulation seems to be less sensitive to the reduction in time step.

We also summarise in table 4 the comparison between DG and Nektar for the pressure drag component, where we see that both methods provide very similar results.

## 4.2 Unsteady wake and comparison to other solvers

In this section we compare the results obtained using the DG formulation, the continuous Spectral code Nektar and the low spatial order Finite Volume (FV) code Fluent. In order to make meaningful comparisons between the different formulations, we estimate the number of DOF for each method ignoring the domain boundaries which moderately reduce the number of DOF.

Taking  $N_{el}$  as the number of triangular elements in the mesh, we approximate the total number of DOF for the DG method as:  $DOF_{DG} \approx \frac{N_{el}}{2}(k+1)(k+2)$ . For the Spectral method, since  $C^0$  continuity is enforced across elements, the required number of DOF is reduced:  $DOF_{Spectral} \approx \frac{N_{el}}{2}k^2$  [19]. For the FV method, which uses piecewise constant functions within each element, a conservative estimate for the number of DOF is  $DOF_{FV} \approx N_{el}$ .

The temporal scheme selected for Fluent is a second order implicit scheme where the pressure-velocity coupling is performed with the PISO algorithm. Spatial discretisation in Fluent is second order upwind for momentum and Presto for pressure.

For the unsteady case with  $Re = 100$ , we compare the Strouhal number,  $St = fb/U_\infty$ , where  $f$  is the vortex shedding frequency computed from the lift force,  $b = 1$  the length

$k$	Nektar	DG	% difference from Nektar
2	1.0523E+00	1.0520E+00	2.44E-02%
3	1.0673E+00	1.0673E+00	1.95E-03%
4	1.0728E+00	1.0728E+00	5.43E-03%
5	1.0785E+00	1.0785E+00	-1.95E-03%
7	1.0858E+00	1.0859E+00	-3.47E-03%
8	1.0880E+00	1.0881E+00	-2.06E-03%

Table 4: Results for DG and Nektar for the pressure component of the drag over the square section cylinder at  $Re=10$ .

of the cylinder and  $U_\infty = 1$  the freestream velocity. We note that for this unsteady case, the viscous component is orders of magnitude smaller than the pressure component and hence the formulation used to compute the viscous forces has little influence. Fig. 8.a depicts the results obtained for the variation of  $St$  number against the number DOF, for the three solvers together with published results from the Nektar code by Darekar and Sherwin<sup>[6]</sup>. We observe that both Nektar and DG give very similar results, providing an accurate solution even for low numbers of DOF. The number of DOF necessary for Fluent to reach comparable accuracy is three times larger than for the two high order methods considered. For completeness, we show the mean drag and RMS lift comparisons using the three codes in Fig. 8.b and 8.c.

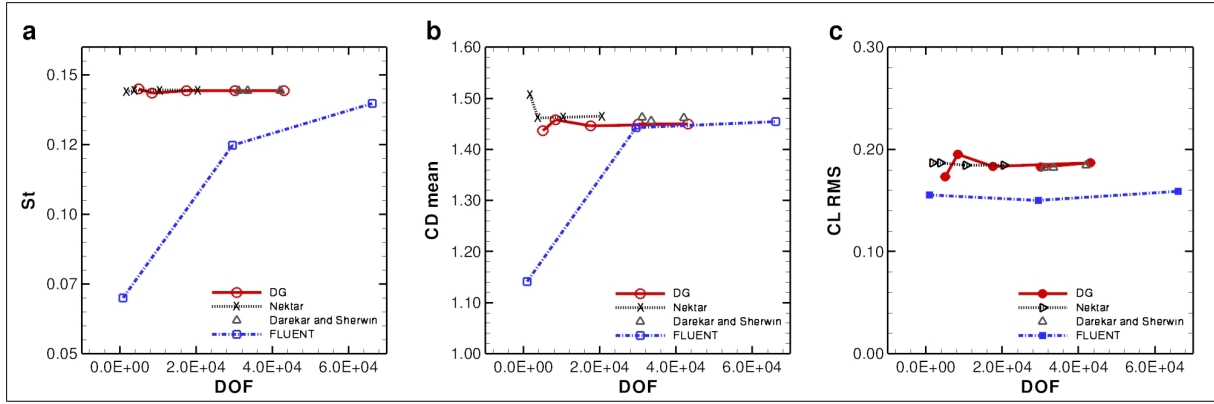


Figure 8: Square section cylinder at  $Re=100$ : (a) Strouhal number derived from lift coefficient, (b) mean drag coefficient and (c) RMS lift force coefficient computed with the DG code with SIPG flux, the continuous Spectral code Nektar and Finite Volume code Fluent. Published results<sup>[6]</sup> for Nektar are also included.

## 5 CONCLUDING REMARKS

An incompressible Navier-Stokes (NS) code that uses a dual splitting scheme for time discretisation and a modal Discontinuous Galerkin Interior Penalty (DG-IP) method for spatial discretisation has been implemented, verified and validated. The DG-IP spatial discretisation has been verified and the role of the penalty parameter in the solution accuracy assessed for the Poisson and Helmholtz equations. The computed convergence rates have been found to agree well with previously published theoretical rates for the SIPG and NIPG fluxes and better than predicted convergence rates were observed for the IIPG flux. It has also been shown that all fluxes provide the same level of accuracy for large values of the penalty parameter.

For the unsteady Stokes problem, it has been observed that the error is of the form  $\mathcal{O}(\Delta t^2 + h^{k+1}/\Delta t)$  such that both spatial and temporal refinement are required to reduce the overall error. Hence, for a given spatial discretisation there is a minimum time step,  $\Delta t_{min}$ , below which no further reduction in error can be achieved.

The method for the full NS equations in conjunction with the DG-IP method and the

modal approach selected requires the time step to be between certain bounds: an upper bound provided by the CFL condition and a lower bound for accuracy as observed for the unsteady Stokes problem.

For the unsteady Stokes and Navier-Stokes problems the method is stable with all fluxes when equal order approximation is used for velocity and pressure.

The solver has been validated for flow past a square section cylinder. This showed that the formulation used to compute the viscous forces can lead to noticeable differences and although this is of minor importance for bluff body flows where pressure forces dominate, this may be of more importance for flows around slender bodies, e.g. airfoils. We have also shown that the developed code provides very similar results to the continuous Spectral code Nektar and requires less degrees of freedom (DOF) than the low order FV code Fluent to achieve comparable accuracy for the vortex shedding aft of a square section cylinder. Indeed, the spatial high order DG solver provides low diffusion and dissipation approximations that lead to a more accurate solution using fewer DOF that can be accomplished with a low order code.

The flexibility of the DG method in handling non conformal meshes and local  $h/p$  adaption while providing as accurate results as continuous Spectral codes is to be exploited in future work.

## 6 ACKNOWLEDGEMENTS

The authors would like to thank the financial support of the John Fell OUP fund as well as the RCUK. They wish to express their gratitude to: Prof. S. Sherwin, Prof. E. Süli, Dr. A. Wathen, Prof. P. Houston, and S. Draper for their valuable comments.

## REFERENCES

- [1] J. D. Anderson, Computational Fluid Dynamics: The Basics with Applications, 6th Edition, *McGraw Hill* (1995).
- [2] ANSYS.Inc Fluent, Release 6.3.26. *www.fluent.com*.
- [3] D.N. Arnold, F. Brezzi, B. Cockburn and L.D. Marini, Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems, *SIAM J. Numer. Anal.*, **39**, 1749–1779 (2001).
- [4] M. Breuer, J. Bernsdorf, T. Zeiser, F. Durst, Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume, *Int. J. of Heat and Fluid Flow*, **21**, 186–196 (2000).
- [5] P. Castillo, Performance of Discontinuous Galerkin Methods for Elliptic PDEs, *SIAM J. Sci. Comput.*, **24**, 524–547 (2002).
- [6] R.M. Darekar and S.J. Sherwin, Flow past a square-section cylinder with a wavy stagnation face, *J. Fluid Mech.*, **426**, 263–295 (2001).

- [7] M. O. Deville, P. F. Fischer, E. H. Mund, High-order methods for incompressible fluid flows, *Cambridge Monographs on Applied and Comp. Mathematics* (2002).
- [8] H.C. Elman, D.J. Silvester and A.J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, *Oxford Univ. Press* (2005).
- [9] V. Girault and F. Wheeler, Discontinuous Galerkin Methods, *Comp. Methods in Applied Sciences*, **16**, 3–26 (2008).
- [10] J.L. Guermond, P. Mineev and Jie Shen, An overview of projection methods for incompressible flows, *Comp. Methods in Applied Mech. and Eng.*, **195**, 6011–6045 (2006).
- [11] J.S. Hesthaven and T. Warburton, Nodal discontinuous Galerkin Methods - Algorithms, Analysis, and Applications, *Springer* (2008).
- [12] Intel Math Kernel Library Reference Manual (2008).
- [13] G.E. Karniadakis, M. Israeli and S.A. Orszag, High-order splitting methods for incompressible Navier-Stokes equations, *J. Comp. Phys.* **97**, 414–443 (1991).
- [14] G.E. Karniadakis and S.J. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, 2nd Edition, *Oxford Science Publications* (2005).
- [15] D. Mavriplis, C. Nastase, K. Shahbazi, L. Wang and N. Burgess, Progress in High-Order Discontinuous Galerkin Methods for Aerospace Applications, *47th AIAA Aerospace Sciences Meeting, AIAA-2009-601* (2009).
- [16] B. Riviere, Discontinuous Galerkin Methods for Solving Elliptic and parabolic Equations: Theory and Implementation, *SIAM-Frontiers in applied mathematics* (2008).
- [17] K. Shahbazi, An explicit expression for the penalty parameter of the interior penalty method, *J. Comp. Phys.* **205**, 401–407 (2005).
- [18] K. Shahbazi, P.F. Fischer, and C.R. Ethier, A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations, *J. Comp. Phys.*, **222**, 391–407 (2007).
- [19] P. Vos, S. J. Sherwin and R. M. Kirby, From  $h$  to  $p$  Efficiently: Implementing finite and spectral  $h/p$  element discretisations to achieve optimal performance at low and high order approximations, *Preprint-J. Comp. Phys.* (2009).