Numerical Analysis of the Incompressible Fluid Flow Around a Two-Dimensional Cylinder by High-Order Discontinuous Galerkin Method

Aureo Quintas Garcia, aureo@utfpr.edu.br[1]
Francisco Augusto Aparecido Gomes, franciscogomes@utfpr.edu.br[2]
Mildred Balin Hecke, mildredhecke@gmail.com[3]

[1]Department of Mathematics, Federal University of Technology - Paraná (UTFPR), Via do Conhecimento, Km 1 - CEP 85503-390, Pato Branco - Paraná - Brazil, Phone: +55 46 3220-2553.
[2]Department of Mechanical Engineering, Federal University of Technology - Paraná (UTFPR), Via do Conhecimento, Km 1 - CEP 85503-390, Pato Branco - Paraná - Brazil, Phone: +55 46 3220-2587.
[3]Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Federal University of Paraná - Paraná (UFPR), Av. Cel. Francisco H. dos Santos, 210, Jardim das Américas - CEP 81531-970, Curitiba - Paraná - Brazil, Phone: +55 41 3361-3218.

Abstract: *We present a numerical solution by discontinuous Galerkin method (DG) applied to the classical two-dimensional cylider-channel experiment. This problem consists of laminar viscous incompressible flow around cylinder inside rectangular channel. Our DG formulation was modified by implementation of the more recent outflow boundary condition (BC) treatment, (Dong* et al. *(2014); Braack and Mucha (2014)), that allows to use reduced size computational domains; as oposed to classical long size domains practices. In the current literature there are applications of this new BC to various incompressible test-flow examples in a Spectral and Finite Elements contexts. The major result achieved by them was this new outflow boundary approach, that allow us significantly reduce computational domain size without presenting errors in the physical characteristic of the flow at outflow of the channel. Accordingly, due to its alredy known capabilities and stablished mathematical basis, it is natural issue to ask about the DG behaviour over this all-useful achievment. In this work the DG method performance, equipped with this new boundary condition, is tested. In this case drag and lift cofficients are computed, so the admensional Strouhal number. Conclusions are traced to show the readness of DG to embody this new boundary condition technique.*
Keywords: *High-order Method, Incompressible Navier-Stokes Equations , Discontinuous Galerkin Method, Outflow Boundary Condition.*

1. INTRODUCTION

The discountinuous Galerkin method (DG) is a very desired numerical tool to solve fluid dynamics plroblems, due to its high flexibility in the discretization of complex geometries, stability for convection-dominated problems and *hp*-adaptivity. It can be understood as a superset of the Finite Volume (FVM) and Finite Element Methods (FEM), containing the more suitable characteristics of this methods. Therefore, it can effectively be used in convective-dominant flows and, at the same time, it is able to produce very accurate results using mesh adaptation [*h*] and/or high-order local approximation [*p*]. For more details and a deepening into theory see Cockburn (2003); Shahbazi (2007); Hesthaven and Warburton (2008).

The DG method suffers the same drawbacks of those presented by FEM and FVM, concerning about outflow boundary modeling. That is, often the computational domain needs to be made larger than the physical problem really demands, otherwise keeping the computational domain with a minimally sufficient size so that it contains the points of interest of the flow, the numerical solution of the problem is often affected with varios types of errors Dong *et al.* (2014). Although there are a number of thecniques to handle that issue (Sani and Gresho, 1994), this is an open problem yet.

The ideal situation would be the one where flow tracked properties could be carried of along the flow and taked off the computational domain without cause non-physical effects upstream outflow boundary. Recently Dong *et al* Dong *et al.* (2014) and Braak *et al* Braack and Mucha (2014) have proposed similar ways to computate flow field variables at outflow boundary in such a manner one can make a drastic domain size cuting off without significantly lose of precision and flow modeling quality. Driven by these results, we were led to ask for DG behavior under this new boundary condition.

The outline of the paper is as follows: In Sction 2 we present the setting up of the numerical experiment. The experiment involves laminar flows around a circular cylinder inside a retangular channel. This experiment reproduces a particular result among the many that were obtained from Schäfer and Turek (Schäfer and Turek, 1996). In the Section 3 we give a brief account for our insertion of algorithm presented in Dong *et al.* (2014) into used DG code. This is made mostly in a flowchart-like way. Computations of lift and drag coefficients and Stouhal number are performed in Section 4, for the cylinder-channel flow. There, flows on severely and on not-severely truncated domains are considered and comparison is made aganist the data from Schäfer and Turek (1996). Finally, concluding remarks are drawn in Section 6.

2. Channel-Cylinder Flow

In this section we present the setting up for DG numerical solution of the viscous incompressible Navier-Stokes equations for the consecrated problem of the flow around a two-dimensional circular cylinder in a channel.

For this test case we have an unsteady, incompressible, two-dimensional, newtonian flow. We used as validation test the numerical experiment conducted by Shäfer and Turek Schäfer and Turek (1996). In their work, among others results and based in a number of numerical experiments criteriously conducted by differents authors, Shäfer and Turek founded refined intervals representing a range of maximum values for each of the parameters Cd, Cl and St. Therefore, the left hand side value of a Cd interval means the least maximum value found for parameter Cd, whereas the right hand side value is the greatest one. Figure 1 shows the geometric setup, where $D = 0.1m$, $H = 0.41m$ and homogeneous Dirichlet boundary conditions. At the inflow boundary we choose the condition

$$u(0, y, t) = \frac{4U_m y(H - y)}{H^2}, \qquad v(0, y, t) = 0, \tag{1}$$

where $U_m = 1.5m/s$. Defining Reynolds number by $Re = \frac{\bar{U}D}{\nu}$ with $\bar{U}(t) = \frac{2}{3}u(0, \frac{H}{2}, t)$, we obtain $Re = 100$ for this case. At this Re unsteady flow regimen is guaranted. The outflow boundary condition is the one dveloped in Dong *et al.* (2014). To start the simulation, we choose homogeneous initial condition.

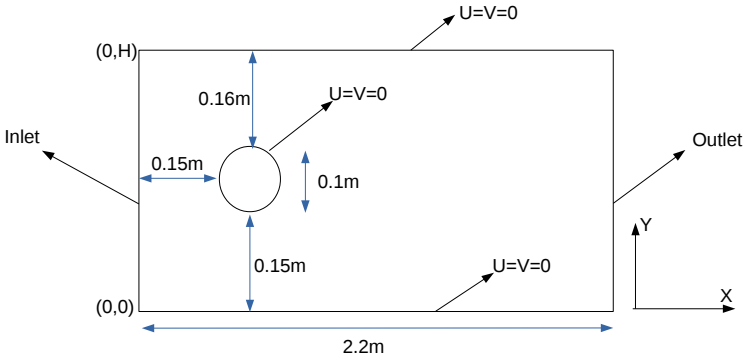

Figure 1: Setup for channel-cylinder experiment. (Free reproduction based in Schäfer and Turek (1996).)

3. Outflow Boundary Conditions

In this work the new outflow bounadry condition, by Dong *et al.* (2014), was used in the DG code to show the effectiveness of DG method equipped with this boundary condition in represent the numerical solutions with great accuracy concerning to the truncated domains. In their work the authors presented, in addition to the Dirichlet and Neumann BC's for pressure and velocity respectively at outflow boundary, the corresponding algorithm to be inserted in a solver for incompressible Navier-Stokes equations. Their paper give also a far complete account of algorithm and insights related to it; so we will not address that point herein. In Braack and Mucha (2014) we can find a more teoretical pont of view as to why this BC design do work. However we present the pressure and velocity formulas and how these are inserted into DG code.

Assuming Ω a two-dimensional open bounded domain, $\partial\Omega = \partial\Omega_D \cup \partial\Omega_o$ a disjoint union where $\partial\Omega_D$ stands for the boundary where velocity $\mathbf{u}$ is prescribed and $\partial\Omega_o$ represents the complementary boundary where neither $\mathbf{u}$ nor p is known, the Navier-Stokes equation can be given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \text{on} \quad \Omega, \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{on} \quad \Omega, \tag{3}$$

where $\mathbf{x}$ is the position vector, $\mathbf{u}(\mathbf{x}, t)$ is the velocity vector, $p(\mathbf{x}, t)$ is pressure field, $f(\mathbf{x}, t)$ is an external field force, $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity and t is the time.

Dong *et al* proposed the following boundary condition on $\partial\Omega_o$

$$\nu \mathbf{n} \cdot \nabla \mathbf{u} - p\mathbf{n} = \mathbf{f}_b(\mathbf{x}, t) + \left[\frac{1}{2}|\mathbf{u}|^2 S_o(\mathbf{n} \cdot \mathbf{u})\right] \mathbf{n}, \quad \text{on} \quad \partial\Omega_o, \tag{4}$$

where $\mathbf{n}$ is the outward unitary normal vector. Observe that, setting ther equation 4 equal to zero, we get the classical do-nothing boundary condition (CDN) Braack and Mucha (2014). Of major importance in this proposal is the $S_o(\mathbf{n} \cdot \mathbf{u})$ term

$$S_o(\mathbf{n} \cdot \mathbf{u}) = \frac{1}{2}\left[1 - tanh\left(\frac{\mathbf{n} \cdot \mathbf{u}}{U_o \delta}\right)\right]. \tag{5}$$

Observe in (5) the dependence of S_o on relative directions by outward normal vector $\mathbf{n}$ and velocity vector $\mathbf{u}$, both evaluated in $\partial\Omega_o$. This dependence led some authors to call this BC "Directional do-Nothing Boundary Condition" (DDN) (Braack and Mucha, 2014). Put simply, this term happens to play its role only when $\mathbf{n} \cdot \mathbf{u} < 0$, that corresponds to the inflow of the computational domain and that action controls grouwth of energy inside it leading to numerical instability or not (Dong *et al.*, 2014).

Let $\mathbf{u}^n$ and p^n denote the velocity and pressure on outflow boundary at time step n. Dong and collaborathors attacked the problem solving the system of equations, composed by Eqs. (2), (3) along with their initial and bounadry conditions, through a strategy that decouples velocity and pressure computations. Based in their strategy and the fact that the used DG code also decouples pressure and velocity, we use the following expression for pressure Dirichlet condition at time step $(n+1)$

$$p^{n+1} = \nu \mathbf{n} \cdot \nabla \mathbf{u}^{n+1} \cdot \mathbf{n} - \frac{1}{2}|\mathbf{u}^{n+1}|^2 S_o(\mathbf{n} \cdot \mathbf{u}^{n+1}) - {\mathbf{f}_b}^{n+1} \cdot \mathbf{n}, \quad \text{on} \quad \partial\Omega_o, \tag{6}$$

while updating in the rate of changing in the velocity components (a Neumann BC) is made by the following equation

$$\mathbf{n} \cdot \nabla \mathbf{u}^{n+1} = \frac{1}{\nu}[p^{n+1}\mathbf{n} - \frac{1}{2}|\mathbf{u}^{n+1}|^2 S_o(\mathbf{n} \cdot \mathbf{u}^{n+1})\mathbf{n} - \nu(\nabla \cdot \mathbf{u}^{n+1})\mathbf{n} + {\mathbf{f}_b}^{n+1}], \tag{7}$$

on $\partial\Omega_o$.

When the flow problem has exact solution, term $\mathbf{f}_b$ comes into play and it is obtained simply replacing the exact values of $\mathbf{u}$ and p into (6) and (7). So it gives computing exactness to outflow boundary; but very often exact flow solutions is not known and $\mathbf{f}_b$ is set to be equal to the zero vector .

While these expressions are almost identical to those in Dong *et al.* (2014), where the only difference stems from a notational issue, the way we have added it into DG code, here, is not the same found there. Aiming to clarify a little more the procedure, we show below a DG code resume and a related block-diagram view.

3.1 DG Code Stucture

The DG code we are developing is built on the top of NUDG code, by J.S. Hesthaven and T. Warburton, freely available in the NUDG website: https://github.com/tcew/nodal-dg/tree /master/Codes1.1 Hesthaven and Warburton (2014). And a higher level outflow boundary condition for incompressible flows, just as the one we are treating herein, constitutes it self one of the parties that we have added to the NUDG code, in order to expand its applicability range. With such a provenance, our DG code inherits the following properties concerning incompressible Navier-Stokes (N-S) equation solver:

1. Temporal splitting scheme: We use stiffly stable timestepping method (Karniadakis *et al.*, 1991). In that time marching scheme each time step is partitioned into three substeps.

   - (1.a) Advection substep.
   - (1.b) Pressure projection.
   - (1.c) Viscous update.

   In the frist substep a simple conservation law is explicitly solved for a first intermediate velocity $\tilde{\mathbf{u}}$ using second-order Adams-Bashforth scheme. Then, in the second substep, a second intermediate velocity $\tilde{\tilde{\mathbf{u}}}$, wich obeys the divergence-free condition, is computed through solving a Poisson problem for pressure, wich is closed with tuned Neumann boundary conditions (Karniadakis *et al.*, 1991). This update the pressure field. By last, a evolution equation relating $\tilde{\tilde{\mathbf{u}}}$ and N-S equation's viscous term, actually a Helmholtz equation, is solved wich updates the velocity field.

2. Spatial discretization: Spatial derivatives for nonlineal advection term in (1.a) is treated a using upwind Discountinuous Galerkin technique. For terms in (1.b) and (1.c), spatial discretization is achieved using an symmetric internal penallization formulation of DG (SIPDG), see Hesthaven and Warburton (2008) for details.

3. Linear sistem solver: It is used a direct approach that consists, in the frist place, of a special permutation of line and columns of linear sistem matrices, for pressure and velocity solutions, that reduces the matrix sparsity pattern and this is accomplished by using the reverse Cuthill-McKee numbering system (George and Liu, 1981). Secondly, taking advantage of system's symetry and its positively definitness (Hesthaven and Warburton, 2008), the solutiuon comes up by Cholesky factorization. Both strategy significantly reduces storage requeriments.

Figure 2 is a schematic view of the algorithm used in this work, where the inclusion of the DDN boundary condition is highlighted.

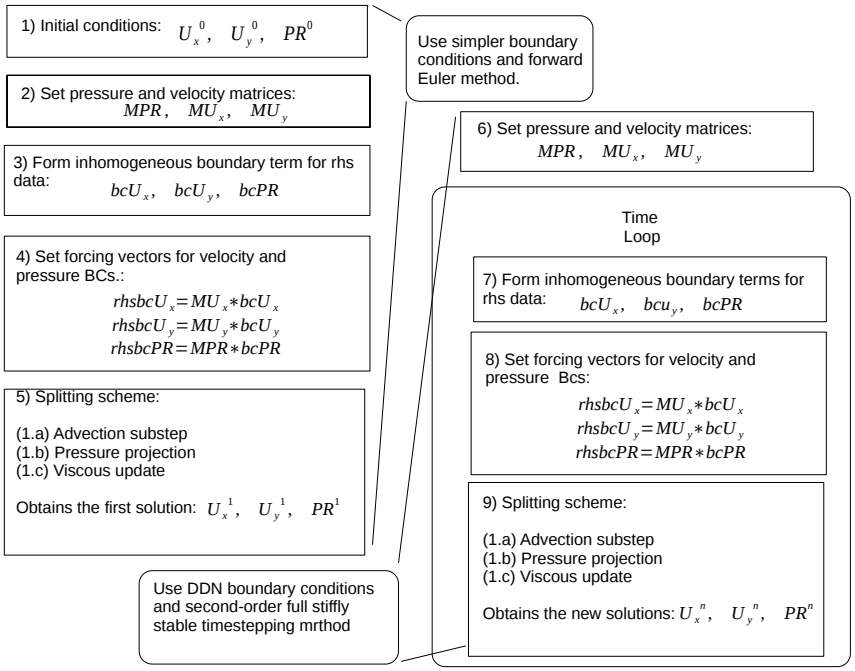

Figure 2: Flowchart of the DG solver.

In this algorithm, through the steps 1 to step 5 we obtain the first solution. Due to the fact that timestepping method is not self-starting, the first time step is a forward Euler method causing pressure and velocity matrices are different from the ones computed in step 6, used from step 7 forward and is generated using the right stiffly stable timestepping method coefficients. In step 7 the inhomogeneous boundary terms are computated using DDN condition at outflow boundary. Inside the time loop, this BC is updatet each time step, to the extent that their formulas receive the corresponding updated $\mathbf{u}^n$ refered in (6) and (7). This algorithmic structure of insertion of the DDN condition is an alternative that we sought to use this new BC applied in a DG implementation.

4. Lift, Drag and Strouhal Number

Here we reproduce, via DG and DDN boundary condition, results found in Schäfer and Turek (1996) whose seting up details are in section 2. Also we set level-set graphics exhibiting the diminished influence, exerted by domain size changing, that the combination of these two numerical techniques provides to end result, even under drastic domain cutting-off.

Figures 4 and 6 shows the level-sets for x-componente of velocity, $\mathbf{U}_x$, for a half-size cutted-off computational domain and a integral one, respectively, being this last according to numerical benchmark in Schäfer and Turek (1996). As one can see, one can not perceive any difference between the two sets for $0 \leq x \leq 1.0$. Figures 3 and 5 shows the respective domain discretizations, while Fig. 7 shows time evolution of C_l coefficient and the associated fast Fourier transform; from which we obtained Stouhal number St.

In the sequel, in complement to the graphical information by level-sets, we present maximum computated values for coefficients: drag, Cd_{max}, lift, Cl_{max} and Strouhal number St, for a sequence of experiments, Table 1. In these experiments , in addition to varying the size of the computational domain and for sake of completeness in testing, we included also, as data, the following test variables: element polynoimial interpolation degree, N, number of elements,

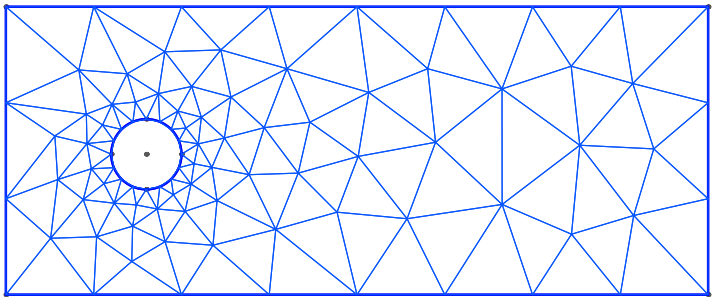

Figure 3: Half-domain discretization. 152 elements, $xmax = 1.0$.

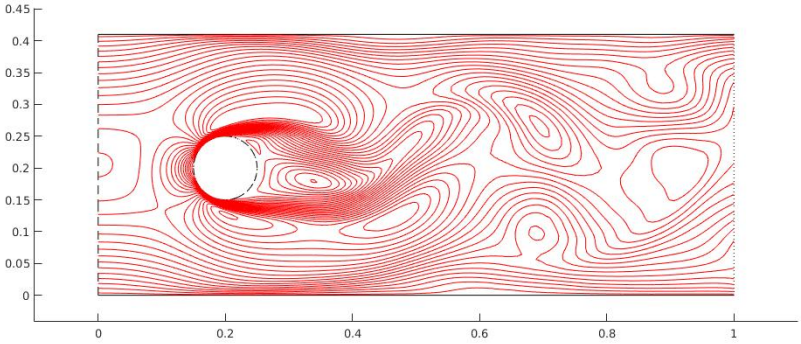

Figure 4: $\mathbf{U}_x$ level-set for a truncated half-domain, at $xmax = 1.0$.

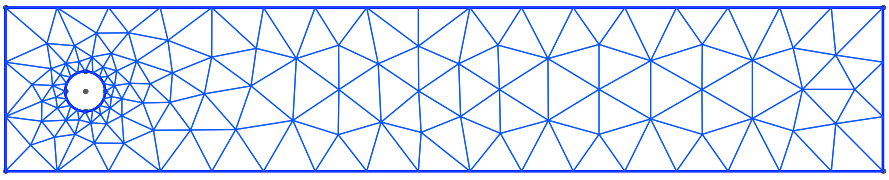

Figure 5: Entire domain discretization. 218 elements, $xmax = 2.2$.

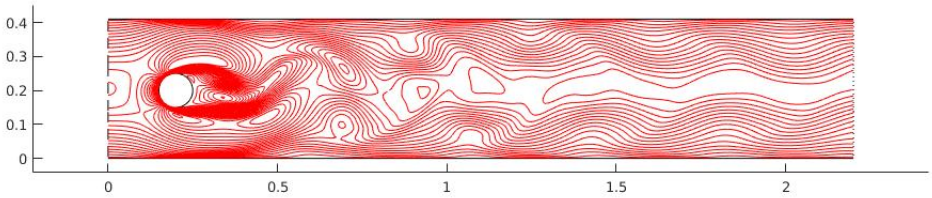

Figure 6: $\mathbf{U}_x$ level-set for entire domain.

final time of program's time-loop (see Fig. 2) and use or not of high-order boundary for cylinder surface (see Hesthaven and Warburton (2008)).

Table 1: N: element interpolation order; Nel: total number of elements; T_f: final time of time loop. (*) our best result for $xmax = 2.2$. (**) result without high-order boundary.

N	Nel	$xmax$	T_f	High order boundary	Cd_{max}	Cl_{max}	St
6	152	1.0	12	yes	3.2492	0.9914	0.3000
6	218	2.2	8	yes	3.2493	0.9765	0.3000
7	152	1.0	8	yes	3.2324	1.0154	0.3000
7	152	1.0	12	yes	3.2324	1.0154	0.3000
7	208	1.0	8	yes	3.2288	1.0260	0.3000
7	208	1.0	12	yes	3.2288	1.0260	0.3000
7*	218	2.2	8	yes	3.2350	1.0074	0.3000
8	152	1.0	8	yes	3.2222	1.0257	0.3000
8	152	1.0	12	yes	3.2222	1.0258	0.3000
8	218	2.2	8	yes	3.2211	1.0196	0.3000
9**	152	1.0	8	no	3.1791	1.0025	0.3000
9	152	1.0	8	yes	3.2254	1.0260	0.3000
10	218	2.2	8	yes	3.2265	1.0207	0.3000

Table 2: Comparison between the computed maximum values intervals, from Schäfer and Turek Schäfer and Turek (1996), and the numerical maximum values obtained by DG code in the present work.

Compared Methods	Higher Maximum			Lower Maximum		
	Cl	Cd	St	Cl	Cd	St
Shäfer & Turek	1.0100	3.2400	0.3050	0.9900	3.2200	0.2950
(DG)	Cl=1.0074		Cd=3.2350		St=0.3000	

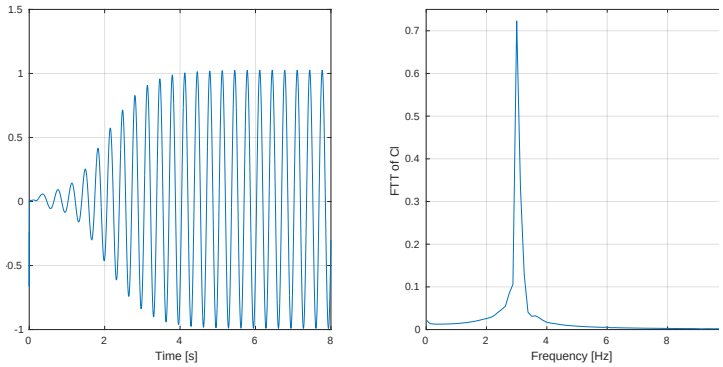

Figure 7: Lift coefficient C_l. 152 elements, $xmax = 1.0$

Results obtained by DG method with DDN boundary condition are in good agreement as in Schäfer and Turek (1996), Tab. 2. For $xmax = 2.2$ our best result (*) is completely inside the bounds settled in Schäfer and Turek (1996) and there is no comparative error. To $xmax = 1.0$ our best result was obtained with $N = 7$, 152 elements, final time equals 8 or 12, with high-order boundary. Also bserve that deviations from benchmark intervals in Tab. 2 fall into only one of the computed parameters, Cd_{max} or Cl_{max}, and that best result deviation occurs only in the third decimal place of computed Cl_{max} value. (Tab. 2, lines 3 and 4). Lastly the absence of a high-order boundary to cylinder surface is felt by undervaluation of Cl_{max} coefficient, while still maintaining the Cd_{max} and St values within the benchmark range stablished in Schäfer and Turek (1996).

5. Concluding Remarks

In our work we incorporated a recent technique proposal to model outflow boundary conditions (by Dong *et al.* (2014) and Braack and Mucha (2014)) into Discountinuous Galerkin Method, aiming a performance analysis. Results demonstrated that incorporation in fact works as flow simulations over severely truncated domains produces nearly equal results to that produced by the ones with integral domains, thats is, long domains. All important as it is for DG method, this save memory allocation and computing time.

Results of our channel-cylinder experiment shows two points that reveals the DG method's good preformance provided with DDN boundary condition. In the frist place, for the long domain case our calculations falls entirely inside Shäfer and Turek intervals (case (*) in Table 1). Secondly, when domain is severely truncated, our calculatios shows that just one parameter falls out of confidence interval as stated by Shäfer and Turek. However out it may be, it still remains close to that interval values.

Finally, the $S_o(\mathbf{n}, \mathbf{u})$ term, as designed by Dong $et\ al$ or that by Braack $et\ al$, constitutes it self an open field for improvements and DG performance surely also depends on it. While first author uses a step-like continuous function (that tends to a step-functuion as δ tends to zero), the second author uses a logical conditional "if" sentence, that is, if $\mathbf{u} \cdot \mathbf{n} < 0$ then $S_o = 0$, else $S_o = 1$. The relative performance of both approaches still deserves a conclusive comparative test.

6. References

Braack, M. and Mucha, P., 2014. "Directional do-nothing condition for the navier-stokes equations". *Journal of Computational Mathematics*, Vol. 32, No. 5, pp. 507–521.

Cockburn, B., 2003. "Discontinuous galerkin methods". In *School of Mathematics*. University of Minnesota.

Dong, S., Karniadakis, G.E. and Chryssostomidis, C., 2014. "A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains". *J. Comput. Phys.*, Vol. 261, pp. 83–105. ISSN 0021-9991. doi:10.1016/j.jcp.2013.12.042. URL `http://dx.doi.org/10.1016/j.jcp.2013.12.042`.

George, A. and Liu, J.W., 1981. *Computer Solution of Large Sparse Positive Definite*. Prentice Hall Professional Technical Reference. ISBN 0131652745.

Hesthaven, J. and Warburton, T., 2008. *Nodal Discontinuous Galerkin Methods - Algorithms, Analysis and Applications*. Springer.

Hesthaven, J. and Warburton, T., 2014. "Nodal dg codes 1.1". https://github.com/tcew/nodal-dg/tree/master/Codes1.1. Accessed: June 30, 2016.

Karniadakis, G.E., Orszag, S.A. and Israeli, M., 1991. "High-order splitting methods for the incompressible Navier-Stokes equations". *Journal of Computational Physics*, Vol. 97, pp. 414–443. doi:10.1016/0021-9991(91)90007-8.

Sani, R.L. and Gresho, P.M., 1994. "Résumé and remarks on the open boundary condition minisymposium". *International Journal for Numerical Methods in Fluids*, Vol. 18, No. 10, pp. 983–1008. ISSN 1097-0363. doi: 10.1002/fld.1650181006. URL `http://dx.doi.org/10.1002/fld.1650181006`.

Schäfer, M. and Turek, S., 1996. "The benchmark problem flow around cylinder ". In H. EH, ed., *Flow Simulation with High-Performance Computers II - Notes on Numerical Fluid Mechanics*. Vol. 52, pp. 547–566.

Shahbazi, K., 2007. *A Parallel High-Order Discontinuous Galerkin Solver for the Unsteady Incompressible Navier-Stokes Equations in Complex Geometries*. Ph.D. thesis, University of Toronto.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.