Non-conforming Elements in Nek5000: Pressure Preconditioning and Parallel Performance

A. Peplinski, N. Offermans, P. F. Fischer, and P. Schlatter

1 Introduction

One of the most important concerns when solving numerically partial differential equations is finding the optimal grid on which the solution will be computed. Unfortunately in most cases it is not an easy task that could be determined in advance without deep understanding of the studied problem. That is why self-adapting algorithms like e.g. adaptive mesh refinement (AMR) have received much attention in past decades and became an important part of many packages for numerical modelling of fluid dynamics e.g. [9, 18]. The goal of AMR is to control the computational error during the simulation by placing higher resolution grids where it is needed. This makes the numerical modelling more robust, and gives the possibility to increase the accuracy of numerical simulations at minimal computational cost. The drawback is, however, increased solver complexity, and it that can have negative effects on the parallel code performance, in particular related to load balancing.

There are number of different AMR schemes, and in the context of the spectral element method (SEM) [16], in which the discretisation is based on a decomposition of the computational domain into a number of non-overlapping, high-order sub-domains called elements, we can distinguish three different categories: The mesh adaptation in this case can mean adjusting the (local) size of an element (r -

A. Peplinski (✉) · N. Offermans · P. Schlatter
Linné FLOW Centre and Swedish e-Science Research Centre (SeRC), KTH Mechanics,
Stockholm, Sweden
e-mail: adam@mech.kth.se; nof@mech.kth.se; pschlatt@mech.kth.se

P. F. Fischer
CS and MechSE Depts., University of Illinois at Urbana–Champaign, Champaign, IL, USA
e-mail: fischerp@illinois.edu

refinement), changing the polynomial order in a particular element (p -refinement), or splitting the element into smaller ones (h -refinement). In this work we concentrate on an h -refinement framework and its implementation in *Nek5000* [8], which is a highly parallel and efficient SEM solver for the incompressible Navier–Stokes equations. In its established version, *Nek5000* only supports conformal elements at constant polynomial order throughout the domain.

The present work was started within EU project CRESTA, where the non-conforming solver for advection-diffusion problem was developed and the basic AMR tasks were implemented using existing external libraries. As h -refinement affects the element connectivity resulting in non-conforming meshes, a special grid manager is required to perform local refinement/coarsening and to build globally consistent meshes. For this task the *p4est* library [1] has been chosen, as it is designed to manipulate domains composed of multiple, non-overlapping logical cubic sub-domains, which can be represented by a recursive tree structure. This library provides element connectivity information for the dual graph, which is later manipulated by ParMETIS [10] producing a new element-to-processor mapping. The final step of grid refinement/coarsening and redistribution is performed within the non-conforming version of *Nek5000*, which utilises the so-called conforming-space/nonconforming-mesh approach based on the previous work of Fischer et al. [7, 11]. As the solver complexity grows special care has been taken to develop efficient tools that can be used within AMR framework. A more detailed description of them and the related scaling tests can be found in [17].

The goal of ExaFLOW is to extend results of CRESTA to the full incompressible Navier–Stokes equations focusing on proper adaptation of the pressure preconditioners for nonconforming SEM. Defining a robust parallel preconditioning strategy has received much attention in past decades, as the linear sub-problem associated with the divergence-free constraint (pressure-Poisson equation) can become very ill-conditioned. In the context of SEM two possible approaches based on the additive overlapping Schwarz method [4, 6] and the hybrid Schwarz-multigrid method [5, 12] were proposed and implemented in *Nek5000*, leading to a significant reduction of pressure iterations.

In the present paper, we discuss the modifications necessary to adapt *Nek5000* for the h -type AMR framework. The article is organised as follows. A short description of SEM and pressure preconditioners is given in Sects. 2 and 3. The following Sects. 4 and 5 describe the algorithmic modifications and parallel performance of the code. Finally, Sect. 6 provides conclusion and future work.

2 SEM Discretisation of the Navier–Stokes Equations

We review briefly the discretisation of the incompressible Navier–Stokes equations to introduce notation and point out algorithm parts that require modification. The more in-depth derivation can be found in e.g. [4]. The temporal discretisation is based on a semi-implicit scheme in which the nonlinear term is treated explicitly

and the remaining unsteady Stokes problem is solved implicitly. To avoid spurious pressure modes our spatial discretisation is based on the $\mathbb{P}_N - \mathbb{P}_{N-2}$ SEM, where velocity and pressure spaces are spanned by Lagrangian interpolants on the Gauss–Lobatto–Legendre (GLL) and Gauss–Legendre (GL) quadrature points, respectively. Note that the basis for velocity is continuous across element interfaces, whereas the basis for pressure is not. Assuming $\mathbf{f}^n$ incorporates all nonlinear and source terms treated explicitly at time t^n , the matrix form of the Stokes problem after applying the Uzawa decoupling reads:

$$\begin{bmatrix} \mathbf{H} & -\frac{\Delta t}{\beta_0} \mathbf{H} \mathbf{B}^{-1} \mathbf{D}^T \\ 0 & \mathbf{E} \end{bmatrix} \begin{pmatrix} \mathbf{u}^n \\ \Delta p \end{pmatrix} = \begin{pmatrix} \mathbf{B} \mathbf{f}^n + \mathbf{D}^T p^{n-1} \\ g \end{pmatrix}, \quad (1)$$

where

$$\mathbf{E} = \frac{\Delta t}{\beta_0} \mathbf{D} \mathbf{B}^{-1} \mathbf{D}^T \quad (2)$$

is the Stokes Schur complement governing the pressure, $\Delta p = p^n - p^{n-1}$ is the pressure update, and g is the inhomogeneity arising from Gaussian elimination. In these equations $\mathbf{H} = -\frac{1}{Re} \mathbf{A} + \frac{\beta_0}{\Delta t} \mathbf{B}$ and $\mathbf{D}$ are the discrete Helmholtz and divergence operators, respectively. β_0 , $\mathbf{A}$ and $\mathbf{B}$ denote here a coefficient from time derivative, a discrete Laplacian and a diagonal mass matrix associated with the velocity mesh. Applying the Uzawa decoupling we use the inverse mass matrix $\mathbf{B}^{-1}$ as approximation of the inverse Helmholtz operator $\mathbf{H}^{-1}$, giving rise to a splitting error. Note that for this splitting method the diagonality of the mass matrix $\mathbf{B}$ is crucial to avoid costly matrix inversion.

All operators $\mathbf{H}$, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{E}$ are symmetric positive definite (SPD) and can be solved with a preconditioned conjugate gradient (PCG) method. Moreover, $\mathbf{E}$ has properties similar to a Poisson operator, and is often referred to as a *consistent Poisson operator*. The systems involving $\mathbf{H}$ and $\mathbf{E}$ are solved iteratively with $\mathbf{E}$ being more challenging, and in the next section we will present the preconditioning strategy for the pressure equation,

$$\mathbf{E} \Delta p = -\mathbf{D} \mathbf{u}. \quad (3)$$

We close this section by shortly presenting the SEM operators. SEM introduces a globally unstructured and locally structured basis by tessellating the domain into K non-overlapping subdomains (deformed quadrilaterals), $\Omega = \bigcup_{k=1}^K \Omega_k$, and representing functions in each subdomain in terms of tensor-product polynomials on a reference subdomain $\hat{\Omega} = [-1, 1]^d$. In this approach every function or operator is represented by its local counterparts, which in case of functions takes the form of a sum over the subdomains

$$f(\mathbf{x}) = \sum_{k=1}^K \sum_i f_i^k h_i(\mathbf{r}).$$

Here, f_i^k and h_i are the nodal values of the function in Ω_k and the base functions in $\hat{\Omega}$, respectively, with i representing the natural ordering of nodes in $\hat{\Omega}$. Combining the coefficients f_i^k one can build global $\underline{f}$ and local $\underline{f}_L$ representations of the function. Each global degree of freedom occurs only once in the global representation, but has multiple copies of faces, edges and vertices related to Ω_k in the local one. To enforce function continuity, the global-to-local mapping is defined as the matrix–vector product $\underline{f}_L = \mathbf{Q}\underline{f}$, where $\mathbf{Q}$ is a binary operator duplicating the basis coefficients in adjoining subdomains. The action $\mathbf{Q}^T \underline{f}_L$ sums multiple contributions to the global degree of freedom from their local values. The assembled global stiffness matrix $\mathbf{A}$ takes the form

$$(\nabla f, \nabla g) = \underline{f}^T \mathbf{A} \underline{g} = \underline{f}^T \mathbf{Q}^T \mathbf{A}_L \mathbf{Q} \underline{g},$$

where a block diagonal matrix $\mathbf{A}_L$ is the unassembled stiffness matrix with each diagonal block consisting of the local stiffness matrix $\mathbf{A}_{ij}^k = \int \frac{dh_i}{dx} \frac{dh_j}{dx} dx$. In practise, the global stiffness matrix is never formed explicitly, and the gather–scatter operator $\mathbf{Q}\mathbf{Q}^T$ is used instead. This operator contains all information about element connectivity.

3 Pressure Preconditioner

An efficient solution of Eq. (3) requires finding an SPD preconditioning matrix $\mathbf{M}^{-1}$ which can be inexpensively applied and which reduces the condition number of $\mathbf{M}^{-1}\mathbf{E}$. Preconditioners based on domain decomposition are a natural choice for SEM as the data is structured within an element but is otherwise unstructured.

An overlapping additive Schwarz preconditioner for Eq. (3) was developed in [4] based on linear finite element discretisation of Poisson operator. It combines solutions of the local Poisson problems in overlapping subdomains $\mathbf{R}_k^T \hat{\mathbf{A}}_k^{-1} \mathbf{R}_k$ with the coarse grid problem $\mathbf{R}_0^T \hat{\mathbf{A}}_0^{-1} \mathbf{R}_0$, which is solved on few degrees of freedom, but covers the entire domain

$$\mathbf{M}^{-1} = \mathbf{R}_0^T \hat{\mathbf{A}}_0^{-1} \mathbf{R}_0 + \sum_k \mathbf{R}_k^T \hat{\mathbf{A}}_k^{-1} \mathbf{R}_k.$$

For the local problems restriction and prolongation operators, $\mathbf{R}_k$ and $\mathbf{R}_k^T$, are Boolean matrices that transfer data to and from the subdomain, and $\hat{\mathbf{A}}_k$ is a local stiffness matrix which can be inverted with e.g. a fast diagonalisation method. Note that action of $\mathbf{R}_k$ and $\mathbf{R}_k^T$ are similar to the gather–scatter operator $\mathbf{Q}\mathbf{Q}^T$.

The coarse grid problem corresponds to the Poisson problem solved on the element vertices only, with $\mathbf{R}_0^T$ being the linear operator interpolating the coarse grid solution onto the tensor product array of GL points. Unlike in [4, 6], $\hat{\mathbf{A}}_0$ is

defined using local SEM-based Neumann operators performing the projection of local stiffness matrices $\mathbf{A}_k$ evaluated on the GLL quadrature points onto the set of coarse base functions b_i representing the linear finite element base on the GLL grid. The coarse base functions are defined in $\hat{\Omega}$ as a tensor-product of the one-dimensional linear functions. The local contribution to $\hat{\mathbf{A}}_0$ is given by $b_i^T \mathbf{A}_k b_j$, and the full $\hat{\mathbf{A}}_0$ is finally assembled by local-to-global mapping summing contributions to the global degree of freedom from their local counterparts. $\hat{\mathbf{A}}_0$ is one of few matrices formed explicitly in *Nek5000*.

On the other hand, the hybrid Schwarz-multigrid preconditioner is based on the multiplicative Schwarz method, which for the two-level scheme takes the form,

$$\mathbf{M}^{-1} = \mathbf{R}_0^T \hat{\mathbf{A}}_0^{-1} \mathbf{R}_0 \left[\sum_k \mathbf{R}_k^T \hat{\mathbf{A}}_k^{-1} \mathbf{R}_k \right],$$

and leads to the following two-level multigrid scheme,

$$\begin{aligned} (i) \quad u^1 &= \sum_k \mathbf{R}_k^T \hat{\mathbf{A}}_k^{-1} \mathbf{R}_k g, \\ (ii) \quad r &= g - \mathbf{A} u^1, \\ (iii) \quad e &= \mathbf{R}_0^T \hat{\mathbf{A}}_0^{-1} \mathbf{R}_0 r, \\ (iv) \quad u &= u^1 + e, \end{aligned}$$

where g , r , e and u are right-hand side, residual, coarse-grid error and solution of equation $\mathbf{A}u = g$, respectively. This method can be extended to a general multilevel solver performing a full V cycle [5, 12]. Notice that by replacing step *ii*) with $r = g$ we obtain the additive Schwarz preconditioner.

4 Adaptation for Non-conforming Meshes

The important advantage of SEM in the context of AMR is its spatial decomposition into elements that can easily be split into smaller ones, and use of the local representation of the operators which decouples intra- and inter-element operations. As h -type AMR using the conforming-space/nonconforming-mesh approach leaves the approximation spaces unchanged, most of the tensor-product operations evaluated element-by-element are preserved, limiting the changes in the algorithm.

The inter-element operations are mostly performed by the gather–scatter operator $\mathbf{Q}\mathbf{Q}^T$ which has to be redefined to include spectral interpolation at the non-conforming faces. Following [7] we consider a non-conforming face shared by one low resolution element (parent) and two (in 3D four) high resolution elements

(children). We introduce a local parent-to-child interpolation operator $\mathbf{J}^{cp}$ which is a spectral interpolation operator with entries

$$(\mathbf{J}^{cp})_{ij} = h_j(\zeta_i^{cp}),$$

where ζ_j^{cp} represents the mapping of GLL points from the child face to its parent. This operator is locally applied to give the desired nodal values on the child face, after $\mathbf{Q}$ copies data from the parent to the children. Building a block-diagonal matrix $\mathbf{J}_L$ with local matrices $\mathbf{J}^{cp}$ one can redefine scatter $\mathbf{J}_L \mathbf{Q}$ and gather–scatter $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^T$ operators, respectively. For more discussion see Fig. 6 and Sect. 4 in [7].

The next crucial modification is diagonalisation of the global mass matrix $\mathbf{Q}^T \mathbf{B}_L \mathbf{Q}$ ($\mathbf{B}_L$ is a block-diagonal built of local mass matrices), whose inverse is required in Eqs. (1) and (2). It is non-diagonal due to the fact that the quadrature points in the elements along the non-conforming faces do not coincide. A diagonalisation procedure is given in [7] and consists of building the global vector $\tilde{\mathbf{b}}$

$$\tilde{\mathbf{b}} := \mathbf{B} \hat{\mathbf{e}} = \mathbf{Q}^T \mathbf{J}_L^T \mathbf{B}_L \hat{\mathbf{e}}_L,$$

and finally setting the lumped mass matrix $\tilde{\mathbf{B}}_{ij} = \delta_{ij} \tilde{b}_i$. $\hat{\mathbf{e}}$ and $\hat{\mathbf{e}}_L$ denote here the global and local vectors containing all ones.

The additive Schwarz preconditioner requires two significant modifications. The first one is related to the assembly of the coarse grid operator $\hat{\mathbf{A}}_0$, which gets more complex for non-conforming meshes. This is due to the fact that the non-conforming mesh introduces hanging vertices located in the middle of faces or edges. These hanging vertices are not global degrees of freedom and cannot be included in $\hat{\mathbf{A}}_0$. To remove them from consideration one has to modify the set of local coarse base functions b_i , which are thus dependent on the shape of the refined region as well as the position and orientation of the child face with respect to the parent one. Unlike the conforming case, where all b_i could be represented by a tensor product of two or three linear functions, the non-conforming mesh requires 5 basic components in two and 21 in three dimensions to assemble all the possible shapes of b_i .

The last missing components are the restriction and prolongation operators, $\mathbf{R}_k$ and $\mathbf{R}_k^T$, for the local Poisson problem. Taking into account the similarity between these operators with $\mathbf{Q} \mathbf{Q}^T$ and following the previous development we use an operator similar to $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^T$, replacing $\mathbf{J}_L$ with the interpolation operator defined on the GL quadrature points. Although this choice seems to be optimal as it preserves properties of the preconditioner and $\mathbf{J}_L^T$ is well defined, our numerical experiments showed a significant increase of pressure iterations in some cases. It was found to be caused by the noise introduced by $\mathbf{J}_L^T$ in the Schwarz operator. To reduce this noise we replaced the transposed interpolation operator with the inverse one, getting a significant reduction of iterations. Unfortunately, such a preconditioner is no longer SPD and PCG cannot be used as an iterative solver in this case. The other problem is the definition of $\mathbf{J}_L^{-1}$, as $\mathbf{J}^{cp}$ can be inverted for

square matrices only, thus excluding p -refinement strategies. To avoid this problem we define a child-to-parent interpolation operator $\mathbf{J}^{pc}$ with the entries

$$(\mathbf{J}^{pc})_{ij} = \begin{cases} h_j(\zeta_i^{pc}) & \text{if } \zeta_j^p \in \partial\Omega^p \cap \partial\Omega^c \\ 0 & \text{otherwise} \end{cases},$$

where $\partial\Omega^p$ and $\partial\Omega^c$ are the parent and child common faces, ζ_j^p is a parent GLL point at the face $\partial\Omega^p$, and ζ_j^{pc} represents the mapping of ζ_j^p to the child face $\partial\Omega^c$. This operator is locally applied to give the desired nodal values on the child face, before $\mathbf{Q}^T$ sums data from the children and the parent. Building block-diagonal matrix $\mathbf{J}_L^{-1}$ consisting of local matrices $\mathbf{J}^{pc}$ one can redefine the gather–scatter $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^{-1}$ operator such that it is appropriate for the pressure preconditioner.

In a similar way we modify the multiplicative Schwarz method, as it shares a number of features with the additive one. In this case we distinguish between Schwarz (acting at single level) and restriction (connecting different levels) operators and apply $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^{-1}$ and $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^T$ to each of them, respectively. Unlike the additive preconditioner, the hybrid one requires also the redefinition of the diagonal weight matrix that indicates the number of sub-domains sharing a given node, and is used to accommodate for overlapping regions. Its value is important as it reduces the largest eigenvalue of the MA operator and defines the smoothing properties of the additive Schwarz step (see [4] and the references therein). In the conforming case its definition is straightforward, however the non-conforming case is more involved as hanging nodes are not real degrees of freedom. In the current implementation the information about node multiplicity on the non-conforming faces is hidden to the parent element, so the parent element sees only one neighbour instead of two (four in 3D). Although this choice gives a preconditioner that significantly reduces the number of pressure iterations, its performance for the studied cases is slightly worse than the performance of the additive Schwarz preconditioner. This can be caused by a non-optimal value of the weight matrix, or by the fact that the hybrid preconditioner is superior over the additive one for high-aspect ratio elements (that are not present in our adaptive simulations).

5 Parallel Performance

The parallel performance test is based on the one of the ExaFLOW flagship calculations, and consists of the turbulent flow around a NACA4412 wing section with 5° angle of attack, at a Reynolds number based on inflow velocity U_∞ and chord length c of $Re_c = 200,000$. It was previously studied in a series of well-resolved large-eddy simulations conducted with the conforming *Nek5000* version, and discussed in detail in [19]. This flow configuration was chosen to illustrate the significant benefit of using AMR, in particular when it comes to the farfield region in the computational domain, but for this article we will only briefly discuss the

Fig. 1 (a) Volume visualisation of that part of the domain covered by refinement levels higher than one for the turbulent flow around a wing profile. The wing vicinity and wake region are resolved and a colour indicates different refinement levels. (b) Strong scaling of the non-conforming *Nek5000* solver for the same case performed on Beskow. The plot shows the time per time step as a function of node number. Each node consists of 32 cores

strong scaling results. We omit here a weak scaling test, as *Nek5000* uses iterative solvers and with the current example we cannot provide meaningful data.

The initial coarse and conforming mesh consisted of 2190 elements with polynomial order $N = 7$ and was evolved for 7.2 time units c/U_∞ to evolve the refinement process using spectral error indicators [13, 14], and allowing for 6 refinements levels. The resulting non-conforming grid was built of 224,272 elements with 76.37×10^6 degrees of freedom, resolving the wing surface and the wake, Fig. 1a. This final mesh was used to test the parallel performance of the non-conforming solver using the petascale Cray XC40 system Beskow at PDC (Stockholm). This system consists of 2060 nodes with 32 cores per node and 2.438 PFlops peak performance. We compare our results with the scaling tests of the conforming *Nek5000* presented in Offermans et al. [15]. The most relevant test in this article is pipe flow at $Re_\tau = 360$ (upper-right plot in their Fig. 5), as it is similar in size with the discussed wing case. We should mention here that our goal is not to improve the parallel performance of the conforming code, but rather to retain it despite of a work imbalance introduced by an additional operator in the direct stiffness summation of the non-conforming solver.

To be able to compare to the conforming solver, we focus on the time evolution loop only, excluding code initialisation, finalisation, mesh rebuilding within AMR and I/O operations. The result of the strong scaling test is presented in Fig. 1b showing the time per time step as a function of node count. This plot is almost identical with the reference one in [15]. Both show slight super-linear scaling between 32 and 256 nodes despite growing work imbalance for the non-conforming solver. We also reach the strong scaling limit at around 256 nodes, which for the conforming solver on Beskow was estimated to be between 30,000 and 50,000 degrees of freedom per core [15]. This shows that the parallel performance of the non-conforming and conforming solvers is almost the same and proves the efficiency of our implementation.

The maximum number of the compute nodes used in the test was not set by the parallel properties of the non-conforming *Nek5000*, but by the quality of the domain partitioning provided by *ParMETIS*. Within ExaFLOW we developed a new grid partitioning scheme for *Nek5000* (not discussed in this paper) that takes into account a core distribution among the nodes, and consists of two steps: inter- and intra-node partitioning. Although this two-level partitioning scheme significantly improves the efficiency of a coarse grid operations for XXT, especially during the setup phase, it relies on the quality of an inter-node partitioning. If the first step gives subdomains with disjoint graphs, the second step cannot be performed. We found that the probability of getting disjoint graphs increases with decreasing number of elements per node, virtually prohibiting the runs with less than 1000 elements per node. However, this limit can differ between simulations. We note however that in the standard production use of the solver this limitation is not critical, as according to [15] it is usually close to the strong scaling limit of conforming *Nek5000*.

6 Conclusions

Within the ExaFLOW project we developed a fully functional SEM-based h -type adaptive mesh refinement (AMR) solver for the incompressible Navier–Stokes equations. This allows for much larger flow cases to be run at reduced cost, as the high resolution grid is placed only in those region where it is needed. At the same time the simulation quality is improved, as the computational error can be controlled during the run.

We have optimised for non-conforming meshes the pressure preconditioners based on the additive overlapping Schwarz and hybrid Schwarz-multigrid methods. To achieve this we modified the base functions for the assembly of a coarse-grid operator to remove hanging nodes, and redefined the direct stiffness summation operator to include spectral interpolation at the non-conforming faces and edges. We introduced two operators $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^T$ and $\mathbf{J}_L \mathbf{Q} \mathbf{Q}^T \mathbf{J}_L^{-1}$ for the different steps in the pressure calculation. The last crucial modification was the diagonalisation of the global mass matrix.

Using real flow cases we show our AMR implementation to be correct and efficient. An important success is the fact that parallel performance of the conforming and non-conforming solvers is very similar, despite the increased complexity of the non-conforming one.

In the future we are going to investigate other definitions of the weight matrix for the hybrid Schwarz-multigrid method, and to test different pressure preconditioners based on the restricted additive Schwarz method [2, 3]. We are going as well to work on the quality of the graph partition, as the two-level partitioning would not accept disjoint graphs on the node's subdomain.

Acknowledgements We would like to thank Niclas Jansson for sharing his expertise on adaptive mesh refinement. The work presented in this publication was supported by a European Commission Horizon 2020 project grant entitled “ExaFLOW: Enabling Exascale Fluid Dynamics Simulation” (grant reference 671571). Computer time was provided by Swedish National Infrastructure for Computing (SNIC) at PDC (KTH Stockholm) and by ExaFLOW at HLRS Stuttgart.

References

1. Burstedde, C., Wilcox, L., Ghattas, O.: *p4est*: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J. Sci. Comput.* **33**(3), 1103–1133 (2011)
2. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.* **21**(2), 792–797 (1999)
3. Efsthathiou, E., Gander, M.J.: Why restricted additive Schwarz converges faster than additive Schwarz. *BIT Numer. Math.* **43**(5), 945–959 (2003)
4. Fischer, P.F.: An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations. *J. Comput. Phys.* **133**, 84–101 (1997)
5. Fischer, P.F., Lottes, J.W.: *Hybrid Schwarz-Multigrid Methods for the Spectral Element Method: Extensions to Navier-Stokes*, pp. 35–49. Springer, Berlin (2005)
6. Fischer, P., Miller, N., Tufo, H.: An overlapping Schwarz method for spectral element simulation of three-dimensional incompressible flow. In: *Parallel Solution of Partial Differential Equations*, pp. 159–180. Springer, New York (2000)
7. Fischer, P.F., Kruse, G.W., Loth, F.: Spectral element methods for transitional flows in complex geometries. *J. Sci. Comput.* **17**(1–4), 81–98 (2002)
8. Fischer, P.F., Lottes, J.W., Kerkemeier, S.G.: Nek5000 Web page (2008). <http://nek5000.mcs.anl.gov>
9. Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J.W., Tufo, H.: FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophys. J. Suppl. Ser.* **131**, 273–334 (2000)
10. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20**(1), 359–392 (1998)
11. Kruse, G.W.: Parallel nonconforming spectral element solution of the incompressible Navier–Stokes equations in three dimensions. Ph.D. Thesis, Brown University, Providence, 1997. UMI Order No. GAX97-38573
12. Lottes, J.W., Fischer, P.F.: Hybrid multigrid/Schwarz algorithms for the spectral element method. *J. Sci. Comput.* **24**(1), 45–78 (2005)
13. Mavriplis, C.: A posteriori error estimators for adaptive spectral element techniques. In: Weseling, P. (ed.) *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics. Notes on Numerical Fluid Mechanics*, pp. 333–342 (1990)
14. Offermans, N.: Towards adaptive mesh refinement in Nek5000. Licentiate Thesis, KTH, Mechanics, 2017
15. Offermans, N., Marin, O., Schanen, M., Gong, J., Fischer, P., Schlatter, P.: On the strong scaling of the spectral element solver Nek5000 on petascale systems. In: *Proceedings of the Exascale Applications and Software Conference 2016*, Stockholm, April 26–29, 2016 (2016)
16. Patera, A.T.: A spectral element method for fluid dynamics: Laminar flow in a channel expansion. *J. Comput. Phys.* **54**(3), 468–488 (1984)
17. Peplinski, A., Fischer, P.F., Schlatter, P.: Parallel performance of h-type adaptive mesh refinement for Nek5000. In: *Proceedings of the Exascale Applications and Software Conference 2016*, Stockholm, April 26–29, 2016, pp. 4:1–4:9 (2016)

18. Rosenberg, D., Fournier, A., Fischer, P., Pouquet, A.: Geophysical-astrophysical spectral-element adaptive refinement (GASpAR): object-oriented h-adaptive fluid dynamics simulation. *J. Comput. Phys.* **215**(1), 59–80 (2006)
19. Vinuesa, R., Negi, P.S., Atzori, M., Hanifi, A., Henningson, D.S., Schlatter, P.: Turbulent boundary layers around wing sections up to $Re_c = 1,000,000$. *Int. J. Heat Fluid Flow* **72**, 86–99 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.