A multiscale numerical algorithm for heat transfer simulation between multidimensional CFD and monodimensional system codes

A Chierici*, L Chirco*, R Da Vià*,†, S Manservisi* and R Scardovelli*

*Department of Industrial Engineering
University of Bologna
Via dei Colli 16, 40136 Bologna (BO), Italy
E-mail: †roberto.davia2@unibo.it

Abstract. Nowadays the rapidly-increasing computational power allows scientists and engineers to perform numerical simulations of complex systems that can involve many scales and several different physical phenomena. In order to perform such simulations, two main strategies can be adopted: one may develop a new numerical code where all the physical phenomena of interest are modelled or one may couple existing validated codes. With the latter option, the creation of a huge and complex numerical code is avoided but efficient methods for data exchange are required since the performance of the simulation is highly influenced by its coupling techniques. In this work we propose a new algorithm that can be used for volume and/or boundary coupling purposes for both multiscale and multiphysics numerical simulations. The proposed algorithm is used for a multiscale simulation involving several CFD domains and monodimensional loops. We adopt the overlapping domain strategy, so the entire flow domain is simulated with the system code. We correct the system code solution by matching averaged inlet and outlet fields located at the boundaries of the CFD domains that overlap parts of the monodimensional loop. In particular we correct pressure losses and enthalpy values with source-sink terms that are imposed in the system code equations. The 1D-CFD coupling is a defective one since the CFD code requires point-wise values on the coupling interfaces and the system code provides only averaged quantities. In particular we impose, as inlet boundary conditions for the CFD domains, the mass flux and the mean enthalpy that are calculated by the system code. With this method the mass balance is preserved at every time step of the simulation. The coupling between consecutive CFD domains is not a defective one since with the proposed algorithm we can interpolate the field solutions on the boundary interfaces. We use the MED data structure as the base structure where all the field operations are performed, allowing the algorithm to be used with all the numerical codes where a MED duplicate of the field solution can be created. The MEDmem libraries come with the Salome platform and include basic methods for handling meshes and fields. We provide results that show the consistency of the proposed algorithm.

1. Introduction
Numerical code coupling allows scientists and engineers to perform simulations of complex systems. These systems can involve several physical phenomena which can occur on different physical scales. When systems with a great number of components are considered, a complete Computational Fluid Dynamic (CFD) simulation cannot be performed because of the its huge computational cost. In order to reduce it, large systems are then usually simulated by using monodimensional codes. If necessary, a better insight of the phenomena occurring in some components of the system can be achieved by coupling a CFD code with the monodimensional
In this way the required computational cost is still acceptable and the solution of the system code can be improved by using the correction of the CFD solution.

In this work we couple a monodimensional code with a CFD code in order to simulate a closed circuit. A sketch of the simulated domain is presented in Figure 1: the circuit contains assemblies of power and cooling units that are connected with an upper and a lower pipe system. The power and cooling units are simulated using the CFD code while the flow inside the pipe system is simulated with the monodimensional code. The numerical coupling of this multiscale simulation can be achieved in two different ways that are represented in Fig. 2. With the decomposed domain defined in Fig. 2 a), the system of interest is divided into several discrete domains that do not overlap and are connected through boundary interfaces. Each domain is simulated with either the system code or the CFD one. The coupling between the different codes is performed via appropriate boundary conditions, as reported in Fig. 3 a). Following the flow direction, the mass flow rate computed on the outlet boundary of the system code is imposed as inlet boundary condition for the CFD code, while the mean pressure calculated on the CFD inlet boundary is imposed as boundary condition on the system code outlet section. A monodimensional domain

Figure 1. Sketch of the simulated system with the power (heater) and cooling (cooler) assembly a) and boundary surfaces where uniform heat flux is imposed b).

Figure 2. Representation of decomposed (a) and overlapping (b) domain coupling strategies for multiscale simulation.
Figure 3. Representation of boundary conditions for numerical code coupling with decomposed (a) and overlapping (b) domain strategies.

that overlaps the CFD domains is instead used with the overlapping domain strategy. In this case the coupling between the two codes is obtained with appropriate boundary conditions when information are passed from the system code to the CFD one. The CFD solution is then used to improve the system code one: this can be performed by calculating some source terms that are then imposed in the system code equations or by modifying some system code parameters [1],[2],[3],[4]. Some performances of domain decomposition and domain overlapping methods are evaluated in [2]. These papers show that the convergence of the decomposed domain strongly depends on the physics of the simulation. In systems driven by an imposed pressure drop on boundary, the mass flow rate is often imposed iteratively from a code to another since the pressure may be discontinuous through the boundary interfaces. Otherwise, with the overlapping domain method the total mass flow rate is continuous across every interface between the monodimensional and CFD domains.

2. Coupling Techniques

In this section we describe the coupling procedure adopted for the multiscale power cooling system simulation. Numerical coupling needs an exchange of information between two or more codes and, in order to perform these operations, interfaces that allow us to access and manipulate input/output solutions are required. For a system code the interface should be able to modify the solution and to access the source terms on every elements at each mesh point of the overlapping monodimensional domain. For a CFD code, the interface may act on boundary points or over points distributed on a sub volume of the domain. This interface should contain a mesh and the solution defined at each mesh point. In the present work, for the realization of the coupling interfaces, we use the MED data format that comes with the SALOMÉ platform. This data structure allows to store numerical fields and meshes within C++ classes. It is then possible to store the required data in the computer memory, without reading and writing operations on external files, obtaining a faster numerical code coupling procedure. The coupling procedure can be divided into 3 main steps: generating the mesh duplicate in the MED format, handling the solution in the data structure and imposing the solution on the target code. From the original solution one need mean integral values (when coupling a CFD and a system code) or field interpolations on a different domain discretization (when coupling two CFD codes). The data transfer is realized by C++ routines designed in an appropriate way for each code which can compute field mean integral values using Gauss integration and point wise field interpolations. In particular, point wise field interpolation routines are based on a search point algorithm, a reverse mapping algorithm (from global coordinate to canonical finite element coordinate system) and an interpolation algorithm based on the classical finite element interpolation. These routines work on MED data format and can be used with all the numerical codes where this format has been adopted for input/output and mesh storage data.
| Property            | Symbol | Value   | Unit     |
|---------------------|--------|---------|----------|
| Viscosity           | $\mu$  | 0.00184 | Pa s     |
| Density             | $\rho$ | 10340  | Kg/m$^3$ |
| Thermal conductivity| $\lambda$ | 10.76896 | W/(m K) |
| Heat specific capacity | $C_p$ | 146     | J/(Kg K) |

Table 1. Physical parameters used in the CFD simulations.

In the case of system code solution correction we indicate with $\lambda$ a variable whose system code solution can be improved with the CFD one. We label with $\lambda_{1D}$ and $\lambda_{CFD}$ the values of $\lambda$ from the system and the CFD code. In the latter case $\lambda_{CFD}$ is calculated as a mean integral value. At each generic time step $n$ a source term $S^n$ is imposed on an element or a point of the system code domain. This source term is calculated as

$$S^n = S^{n-1} - \omega_\lambda (\lambda_{CFD} - \lambda_{1D}),$$

(1)

where $\omega_\lambda$ is an under relaxation parameter smaller than 1. In this way, when $\lambda_{1D} = \lambda_{CFD}$ the source term $S$ does not impose any correction. In our simulation $\lambda$ can be the pressure loss across the CFD domain and across the monodimensional overlapping domain or the fluid enthalpy on boundary interfaces.

In the case we need to correct the CFD code or imposing boundary conditions we need to model a solution profile from a single point value. As described in [5] and references therein, various methods are available. Above all, the imposition of a uniform flat profile is the easiest method since it doesn’t require the solution of an additional equation or a modification of the numerical code. Sometimes in the case of the Navier Stokes equation this method can lead to an overestimation of the pressure losses inside the CFD component as in the region near the inlet boundary strong velocity gradients take place [2].

3. Numerical Modelling

The operative fluid is Lead Bismuth Eutectic (LBE), whose physical properties are reported in Table 1, while in Table 2 the main geometrical parameters of the simulated domain are reported. We remark that this fluid is characterized by a low value of the Prandtl number, in particular $Pr = 0.025$. As the fluid flow is turbulent, we solve the Reynolds Averaged Navier Stokes equations

$$\nabla \cdot \mathbf{u} = 0,$$

(2)

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \nabla \cdot [\mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) - \rho \mathbf{u} \mathbf{u}^T] + \rho \mathbf{g},$$

(3)

$$\rho C_p \left( \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) = \nabla \cdot \left[ \lambda \nabla T - \rho C_p \mathbf{u} T \mathbf{u}^T \right] + Q,$$

(4)

Table 2. Geometrical parameters of the simulated domain.

| 1D pipe   | length | 1.5 m |
|-----------|--------|-------|
|           | radius | 0.01 m|
| Heater    | length | 0.2 m |
|           | inner radius | 0.004 m|
|           | outer radius | 0.0303 m|
| Cooler    | length | 0.2 m |
|           | cooling rods radius | 0.01 m|
|           | width | 0.05 m|
where \( u, P \) and \( T \) are the mean velocity, mean pressure and mean temperature. The unknown Reynolds stresses \( \rho \omega \overline{\omega} \) and the turbulent heat flux \( \rho C_p \overline{\omega T} \) are modeled with the following logarithmic four parameter turbulence model

\[
\frac{\partial K}{\partial t} + u \cdot \nabla K = \nabla \cdot \left[ \left( \nu + \frac{\nu_t}{\sigma_k} \right) \nabla K \right] + \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla \cdot \nabla K + \frac{P_\theta}{eK} - C_\mu e^\omega,
\]

\[
\frac{\partial \Omega}{\partial t} + u \cdot \nabla \Omega = \nabla \cdot \left[ \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla \Omega \right] + 2 \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla K \cdot \nabla \Omega + \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla \Omega \cdot \nabla \Omega,
\]

\[
\frac{\partial K_\theta}{\partial t} + u \cdot \nabla K_\theta = \nabla \cdot \left[ \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla K_\theta \right] + \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla K_\theta \cdot \nabla K_\theta + \frac{P_\theta}{eK_\theta} - C_\mu e^{\Omega_\theta},
\]

\[
\frac{\partial \Omega_\theta}{\partial t} + u \cdot \nabla \Omega_\theta = \nabla \cdot \left[ \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla \Omega_\theta \right] + 2 \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla K_\theta \cdot \nabla \Omega_\theta + \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla \Omega_\theta \cdot \nabla \Omega_\theta
\]

\[
\frac{\partial \omega}{\partial t} + u \cdot \nabla \omega = \nabla \cdot \left[ \left( \nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right] + \left( \nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \cdot \nabla \omega
\]

The variables \( \nu_t \) and \( \alpha_t \) represent the eddy viscosity and the eddy thermal diffusivity, while \( K \) is the natural logarithm of the turbulent kinetic energy \( k \), \( \Omega \) is the natural logarithm of \( k \) specific dissipation \( \omega \), \( K_\theta \) is the natural logarithm of mean squared temperature fluctuations \( k_\theta \) and \( \Omega_\theta \) the natural logarithm of \( k_\theta \) specific dissipation, \( \omega_\theta \). For details on the benefits of a logarithmic turbulence model and on the formulation of the eddy viscosity and eddy thermal diffusivity on can refer to \([6],[7],[8],[9] \) and \([10]\). In the present turbulence model we adopt a realizability constraint where the eddy viscosity is limited from above by a value that depends on the local values of the mean strain rate tensor modulus and of the turbulent kinetic energy \( k \) \([11],[12]\). The system of equations \((2-8)\) is solved with an in-house finite element code. The equations are upwinded with a SUPG formulation \([13]\). In the system \((5-8)\), the cross diffusion terms

\[
\left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla K \cdot \nabla K, \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla \Omega \cdot \nabla \Omega, \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla \Omega \cdot \nabla \Omega,
\]

\[
\left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla K_\theta \cdot \nabla K_\theta, \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla K_\theta \cdot \nabla \Omega_\theta, \left( \alpha + \frac{\alpha_t}{\sigma_\theta} \right) \nabla \Omega_\theta \cdot \nabla \Omega_\theta
\]

are seen as additional convective terms. We introduce the fictitious velocities \( u_K, u_\Omega, u_{K_\theta}, u_{\Omega_\theta} \) and rewrite the cross diffusion terms as

\[
u K \cdot \nabla K, \ u_\Omega \cdot \nabla \Omega, \ u_{K_\theta} \cdot \nabla K_\theta, \ u_{\Omega_\theta} \cdot \nabla \Omega_\theta
\]

For the system \((5-8)\) the SUPG upwind terms are calculated using the modified velocity field \( \tilde{u} = u - u_i \), with \( i=K, \Omega, K_\theta, \Omega_\theta \). In this way some diffusion is added also in the direction of sharp gradients of the turbulence variables, which means near the solid boundaries. In the domain regions where the turbulence variable gradients are small, if compared to the velocity field, the SUPG formulation is equivalent to the standard one.

### 4. Results

In this section we show the results we obtained from the simulation of the multiscale system.

The power cooling unit consists of a cylindrical annulus, where a uniform heat flux is applied on the inner surface, a cooling device, an additional heat exchanger, a volume component, a pump and a pipe system. The cylindrical annulus, hereafter labeled as \( \text{heater} \) component, and
Figure 4. Schematic representation of the transient simulation.

the cooling device, labeled as *cooler* component, are simulated with the CFD code and with an equivalent pipe component in the system code. The additional heat exchanger and the pump are modelled as point wise components within the system code simulation. We simulate a transient situation as reported in Fig. 4, where \( N \) indicates the time step. The simulation starts from a stationary solution of the CFD problem. For the first \( N_{\text{start}} \) time steps the system code reaches a stationary solution based on the stationary CFD one, i.e. correcting the pressure losses and the enthalpy values. From iteration \( N_{\text{start}} \) we couple the solution of the CFD and system codes and reach a new stationary solution. At time step \( N_{\Delta P} \) we increase the pressure gain on the pump component. The simulation is performed for four different values of the under relaxation parameters, namely \( \omega_p \) for the pressure losses and \( \omega_T \) for the fluid enthalpy, that are 0.01, 0.05, 0.2 and 0.5. The system is driven by the pressure gain imposed on the pump component. For the CFD components, on the inlet boundaries we impose the mass flow rate and the temperature that are read from the system code solution on the boundaries interfaces. For the turbulence variables we impose uniform values calculated as

\[
K_{\text{in}} = \ln(1.5(Iv_{\text{mid}})^2), \quad \Omega_{\text{in}} = \ln\left(\frac{e^{K_{\text{in}}}}{L}\right)
\]

\[
K_{\theta,\text{in}} = K_{\text{in}} - \ln(Pr), \quad \Omega_{\theta,\text{in}} = \Omega_{\text{in}} - \ln(Pr),
\]

where \( I \) is the turbulence intensity, \( I = 0.16Re_h^{1/8} \), and \( L \) is a characteristic length, \( L = 0.07D_h \). \( Re_h \) and \( D_h \) represent the Reynolds number and the hydraulic diameter, while \( v_{\text{mid}} \) is the averaged velocity component in the normal direction to boundary interface. On wall surfaces, for the turbulent variables we set boundary conditions in accordance with their near wall behavior [7], while on the outlet boundaries we impose homogeneous Neumann boundary conditions. On the boundaries \( \Gamma_h \) and \( \Gamma_c \), as reported in Fig. 1, we impose a uniform heat flux so that, in stationary condition, the temperature difference between the mean values on the heater inlet and the cooler outlet is zero. In order to test the field interpolation functions, the simulations are performed considering the heater as a whole component or as split in two regions, labeled by HEAT1 and HEAT2, characterized by a different domain discretization. The latter case leads to a domain decomposition coupling method, when passing information from the first half of the heater component to the consecutive one. In the following, when
dealing with the heater component split in two parts, we will refer to $\Gamma_{\text{in}}^1$, $\Gamma_{\text{out}}^1$, $\Gamma_{\text{in}}^2$, $\Gamma_{\text{out}}^2$ as the inlet and outlet boundary sections of HEAT1 and HEAT2. As the fluid physical properties do not depend on the pressure field, we can choose whether to perform or not a backward pressure field interpolation from $\Gamma_{\text{in}}^2$ to $\Gamma_{\text{out}}^1$, as represented in Fig. 5. In fact we need to calculate the overall pressure drop across the heater component. If we perform the backward pressure field interpolation then $\Delta P_{\text{CFD}} = P_{\Gamma_{\text{in}}^1} - P_{\Gamma_{\text{out}}^2}$, otherwise $\Delta P_{\text{CFD}} = (P_{\Gamma_{\text{in}}^1} - P_{\Gamma_{\text{out}}^1}) + (P_{\Gamma_{\text{in}}^2} - P_{\Gamma_{\text{out}}^2}) = \Delta P_{\text{HEAT}1} + \Delta P_{\text{HEAT}2}$, where on the outlet boundaries $\Gamma_{\text{out}}^1$ and $\Gamma_{\text{out}}^2$ an arbitrary pressure value can be set as boundary condition. The backward pressure field interpolation can be avoided when a uniform pressure field can be imposed on the outlet boundary, for example in the case of fully developed flow. In Fig. 6 the velocity and temperature fields on the outlet sections of HEAT1 are reported together with the interpolated values that are imposed as inlet boundary conditions on the inlet section of HEAT2. In order to estimate the interpolation field accuracy we report in Table 3 the mean velocity and bulk temperature values of both original and interpolated fields. We also report the fractional difference between the original and the interpolated values, namely $\Delta v_m,\text{frac}\% = (v_{m,\text{HEAT}1} - v_{m,\text{HEAT}2})/v_{m,\text{HEAT}1}$ and $\Delta T_{b,\text{frac}\%} = (T_{b,\text{HEAT}1} - T_{b,\text{HEAT}2})/T_{b,\text{HEAT}1}$. For the velocity field interpolation the difference $\Delta v_m,\text{frac}\%$ is small ($<0.25\%$), while for the temperature field interpolation the
difference is even smaller (< 0.01%). For the case of split heater component we use a domain decomposition coupling method between HEAT1 and HEAT2 with and without backward pressure field interpolation. In the following, we will label with case 1 the heater component as a whole component, with case 2 the simulations with the heater split into two parts and with backward pressure field projection, while with case 3 the simulations with the heater component split into two parts but without the pressure coupling between HEAT1 and HEAT2. In Fig. 7

![Graph](image-url)

**Figure 7.** Behavior of mean temperature a), on the outlet boundary of the heater component, and of pressure losses b) across the heater component, as a function of the step count.

| Case 1 | Case 2 | Case 3 |
|--------|--------|--------|
| Mass flow rate [kg/s] | 3.2496 | 3.2502 | 3.2502 |

we report a comparison of the CFD and system code results of case 1, for the heater component. The comparison is performed between the values obtained with the four different values of both $\omega_P$ and $\omega_T$, as a function of the step count. In particular, in Fig 7 a) we compare the mean integral temperature calculated on the outlet boundary of the heater component with the point
The wise value coming from the system code solution. We can see that as the under relaxation parameter increases the convergence between CFD and system code solutions is reached with a smaller number of iterations. A stationary value is reached after the coupling of CFD and 1D codes, before changing the value of the pressure gain on the 0D pump element, and also after the change of $\Delta P$. The value $\omega_T = 0.01$ leads to a slow numerical convergence between the CFD and 1D solutions. In order to evaluate the convergence of the pressure losses, in Fig. 7 b) we show the behavior of the parameter $\Delta P_{frac}$, $\Delta P_{frac} = |\Delta P_{CFD} - \Delta P_{1D}|/\Delta P_{CFD}$, for the different values of $\omega_p$. As for the temperature case, the convergence between the values of pressure losses across the CFD domain, $\Delta P_{CFD}$, and the ones across the monodimensional domain, $\Delta P_{1D}$, is faster as the under relaxation parameter increase. As a final comparison between case 1, case 2 and case 3, we report in Table 4 the values of the stationary mass flow rate obtained after the change of the pump pressure gain. We can see that there is a slight difference between the value obtained with the heater as a whole component and the values obtained splitting the heater in the two components HEAT1 and HEAT2. In particular the mass flow rates for case 2 and case 3 are the same, which means that, for this particular simulated system, a uniform pressure value can be set as outlet boundary condition on HEAT1 without altering the final result. In Fig. 8 we report the stationary turbulence variable fields of the cooler component. In particular, in Fig. 8 a), the streamlines are reported with the field of the logarithmic value of turbulent kinetic energy specific dissipation rate, $\Omega$. We can observe the presence of some
recirculating regions behind the cooling surfaces. These regions are also characterized by a large value of the logarithmic mean squared temperature fluctuations $K_\theta$. The use of the turbulence limiters proposed in [11] increased the solution stability of the dynamical turbulence system of equations, preventing anomalous growth of the turbulence variables in the recirculation regions.

5. Conclusions
In the present work we presented a coupling algorithm that can be used to perform both multiscale and multiphysics simulations. The proposed algorithm has been tested simulating a complex system, involving several CFD domains and a closed monodimensional pipe system. Both the overlapping and decomposed domain methods have been used in order to realize, respectively, a coupling between a system code and a CFD one and a coupling between two consecutive CFD domains. The data exchange between different codes requires the creation of proper interfaces that are used to access and manipulate code solutions. We used the MED data format, that comes with the MEDmem libraries, for the creation of these interfaces, and the code coupling procedure is realized without using reading/writing procedures on external files. In this way, by using the MEDmem libraries, the coupling operations are performed within the computer memory, allowing a faster numerical code coupling execution. The system of interest has been simulated with various configurations, in order to test both the overlapping and the decomposed domain methods. In particular, for the overlapping domain method, we tested four different values of the under relaxation parameters used to correct the pressure losses and the temperature values of the system code solution. Within the tested range of parameters values, the convergence between the system code solution and the CFD one was found to be faster as the under relaxation parameters increase, without stability problems. For the decomposed domain method we focused our attention on a CFD component and considered it as a whole component and as two different components connected through a boundary interface. The results obtained with the CFD component split in two parts showed only a slight difference with respect to those obtained with the CFD component as a whole component. We can conclude that the proposed algorithm can be used to perform multiscale and multiphysics simulations.

References
[1] Cerroni D 2015 Multiscale multiphysics coupling on a finite element platform Ph.D. thesis University of Bologna
[2] Grunloh T P and Manera A 2016 Annals of Nuclear Energy 90 422–32
[3] Bavi`ere R, Tauveron N, Perdu F, Garré E and Li S 2014 Nucl. Eng. Des. 277 124–37
[4] Tenchine D et al. 2012 Nucl. Eng. Des. 245 140–52
[5] Prince J A 2015 Coupled 1D-3D simulation of flow in subway transit networks Ph.D. thesis Imperial College London
[6] Ilinca F, Hetu J F and Pelletier D 1998 Computers and Fluids 27 291–310
[7] Da Vià R, Manservisi S and Menghini F 2016 Int. J. Heat Mass Tran. 101 1030–41
[8] Manservisi S and Menghini F 2014 Int. J. Heat Mass Tran. 69 312–26
[9] Manservisi S and Menghini F 2014 Nucl. Eng. Des. 273 251–70
[10] Cerroni D, Da Vià R, Manservisi S, Menghini F, Pozzetti G and Scardovelli R 2015 J. of Phys.: Conf. Series 655 012046
[11] Park C H and Park S O 2005 Int. J. Comput. Fluid D. 19 79–86
[12] Durbin P A 1996 Int J. Heat Fluid Fl. 17 89–90
[13] Bochev P B, Gunzburger M D and Shadid J N 2004 Comput. Methods in Appl. Mech. Eng. 193 2301–23