Journal of Computational Physics 228 (2009) 3708–3741

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A composite grid solver for conjugate heat transfer in fluid–structure systems

William D. Henshaw [a,*], Kyle K. Chand [b]

[a] Centre for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States
[b] Science and Technology Computing Division, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States

ARTICLE INFO

Article history:
Received 21 October 2008
Received in revised form 26 January 2009
Accepted 3 February 2009
Available online 20 February 2009

Keywords:
Conjugate heat transfer
Overlapping grids
Incompressible flow
Numerical methods
Multi-domain solvers

ABSTRACT

We describe a numerical method for modeling temperature-dependent fluid flow coupled to heat transfer in solids. This approach to conjugate heat transfer can be used to compute transient and steady state solutions to a wide range of fluid–solid systems in complex two- and three-dimensional geometry. Fluids are modeled with the temperature-dependent incompressible Navier–Stokes equations using the Boussinesq approximation. Solids with heat transfer are modeled with the heat equation. Appropriate interface equations are applied to couple the solutions across different domains. The computational region is divided into a number of sub-domains corresponding to fluid domains and solid domains. There may be multiple fluid domains and multiple solid domains. Each fluid or solid sub-domain is discretized with an overlapping grid. The entire region is associated with a composite grid which is the union of the overlapping grids for the sub-domains. A different physics solver (fluid solver or solid solver) is associated with each sub-domain. A higher-level multi-domain solver manages the entire solution process.

We propose and analyze some centered discrete approximations to the interface equations that have some desirable stability properties. The coupled interface equations may be solved directly when using explicit time-stepping methods in the sub-domains, resulting in a strongly coupled approach. The stability of the interface treatment in this case is independent of the relative sizes of the material properties in the two domains with the time-step only depending on the usual von Neumann conditions for each sub-domain. For implicit time-stepping methods we solve the interface equations in a weakly coupled fashion to avoid forming a coupled implicit system across all sub-domains. The convergence of this approach does depend on the relative sizes of the thermal conductivities and diffusivities. We analyze different iteration strategies for solving these implicit equations including the use of mixed (Robin) approximations at the interface.

Numerical results are presented to illustrate the method. The accuracy of the technique is verified using the method of analytic solutions and by computing the solution to some heat exchanger problems where the exact solution is known. The technique is also applied to the modeling of an inertial-confinement-fusion hohlraum target and the flow of coolant past an hexagonal array of heated fuel rods. The multi-domain solver runs in parallel on distributed memory computers and some parallel results are provided.

* Corresponding author. Tel.: +1 925 423 2697.
E-mail addresses: henshaw1@llnl.gov (W.D. Henshaw), chand1@llnl.gov (K.K. Chand).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.02.007

1. Introduction

There are many interesting scientific and engineering problems that involve the coupling of fluid flow to heat transfer in solids. These include modeling of heat exchangers, cooling of turbine blades in jet engines, nuclear reactors and cooling of computer components to name a few. This manuscript outlines a first step towards the development of a framework and numerical approximations for simulation of some of these important applications. We describe a flexible approach for modeling heat transfer in fluid–solid systems based on the use of composite overlapping grids. The approach uses different physics solvers in the different fluid and solid domains. The solutions are coupled at the fluid–solid interfaces using the continuity of temperature and heat flux. There can be any number of different fluid or solid domains and any number of different physics solvers. The fluids are modeled using the incompressible Navier–Stokes equations with the Boussinesq approximation. The solids are modeled with the heat equation. Each fluid or solid domain is discretized with an overlapping grid. Curvilinear boundary fitted grids are used at boundaries and interfaces. The entire domain is represented with a composite grid that holds the union of the sub-domain overlapping grids. The primary goals of the work presented here are to develop the multi-domain numerical method and computational framework, to use mathematical and numerical analysis to understand properties of the coupled approach, and to verify the accuracy of the technique. Simulation of many realistic problems will require extensions of the present work, such as the addition of appropriate turbulence models, but this is left to future work.

The multi-domain solution of all fluid and solid domains is advanced in time in a domain-split, weakly coupled, manner. During each composite time-step, the solution in each sub-domain is advanced with an explicit or implicit predictor–corrector time-stepping algorithm using the physics solver for that domain. The interface equations are updated after the predictor and corrector steps. When explicit time-stepping is used, the interface equations are solved as a separate set of coupled equations, resulting in an effectively strongly coupled algorithm. We propose a non-standard *centered* interface approximation. An analysis shows that with the centered interface approximation, the stability of the scheme is independent of the relative sizes of thermal conductivities and thermal diffusivities in the adjacent domains. The time-step for the overall scheme is no worse than the time-step determined for the individual sub-domains. This is in contrast to the more commonly used approach [1]. When implicit time-stepping algorithms are used in the sub-domains, the interface equations are solved in a segregated fashion using a Dirichlet condition on one side and a Neumann condition on the opposite side of the interface. In this case it may be necessary to iterate and use additional corrector steps in order to satisfy the coupled interface conditions to the desired tolerance. We describe and analyze an iteration strategy for solving these coupled implicit systems. The iteration converges rapidly when a certain ratio involving the thermal conductivities and diffusivities of adjacent domains is small (or large). The convergence of the iteration can be accelerated using a relaxation parameter. We also analyze the use of a mixed (Robin) interface condition instead of the standard Dirichlet–Neumann approach and show that it has attractive convergence properties especially for the situation when the Dirichlet–Neumann approach has difficulties.

Our multi-domain approach is based on the use of overlapping grids. This method, as discussed in Chesshire and Henshaw [2], allows complex domains to be represented with smooth structured grids that can be aligned with the boundaries. Compared to a multi-block grid, it is easier to construct an overlapping grid for a complex domain since the component grids are not constrained to match exactly. The use of smooth grids is important for obtaining accurate answers especially when using high-order accurate methods. Boundary fitted grids are important for accurate implementation of boundary conditions and for representing boundary layer phenomena. The use of structured grids is important for performance and low memory use. Moreover, since the majority of an overlapping grid often consists of Cartesian grid cells, the speed and low memory requirements inherent with Cartesian grids can be substantially retained. The overlapping grid technique is especially attractive for handling problems with moving or deforming boundaries since the grids remain smooth and can be rapidly generated. Although the usual interpolation used at overlapping grid interfaces is not conservative, conservative interpolation for overlapping grids can be constructed [3]. However, in our experience and in the work of many others, the simpler non-conservative interpolation has worked well, even for very difficult problems involving strong shocks and detonations [4,5].

Overlapping grids have been used to solve a wide class of problems efficiently and accurately. The first use of overlapping grids (called *composite grids* at the time) appeared in papers by Volkov [6,7], who considered approximations to Poisson's equation in regions with corners. Other pioneering work includes that of Starius [8–10], Kreiss [11] and Steger et al. [12] who referred to the approach as *Chimera* grids. Since this early work, the overlapping grid technique has been used successfully to solve a wide variety of problems in high-speed reactive flow [4,5,13,14], reactive and non-reactive multi-material flow [15,16], combustion [17], aerodynamics [18–24], blood flow [25], electromagnetics [26], flows around ships [27], visco-elastic flows [28] and flows with deforming boundaries [29–31], among others. We also note that a hybrid scheme using overlapping grids in fluid regions and unstructured grids in solid regions has been applied to some conjugate heat transfer problems [32]. However, to our knowledge, the present work is the first application of composite grids to conjugate heat transfer problems where both fluids and solids are solved using overlapping grids.

There are a variety of strategies that have been used to solve conjugate heat transfer problems. Finite difference, finite-volume, finite-element, boundary-element and spectral-element approximations have all been applied [33–39]. The different multi-domain strategies are distinguished by the degree to which domains are coupled. In the strongly coupled approach, a single large monolithic system is defined for the entire composite domain. This approach is often the most robust. A common technique for conjugate heat transfer problems is to solve the fluid equations for the velocities and tem-

perature in the entire domain, but force the velocities to be zero or small in the solid regions [33,34]. In the weakly coupled approach, separate solution algorithms are used in different domains with solutions coupled at the interface [40]. This approach has some practical advantages in terms of re-use of existing physics codes without the need to develop a new monolithic approximation. There are yet other approaches that lie somewhere between the strongly coupled and weakly coupled. Some practitioners, for example, solve for a single temperature equation across all domains while having separate fluid solvers in different domains [35,36].

The stability of the segregated interface approach for coupled heat equations has been studied by Giles [1]. He analyzed a particular discretization and showed that for stability reasons the fluid domain should generally be given the Dirichlet condition for continuity of the temperature and the solid domain the Neumann condition for continuity of the heat flux. Giles also showed that the time-step restriction of the coupled problem was sometimes smaller than those from the sub-domains and depended on the relative sizes of the heat capacities and grid spacings in the sub-domains. Roe et al. [40] considered a different discrete approximation to the interface equations that improved the stability characteristics. There has also been much work on solution strategies for domain-decomposition problems and many of these ideas are applicable to conjugate heat transfer problems, see for example [41–43].

An outline of the paper now follows. In Section 2, we define the problem to be solved in terms of the partial differential equations (PDEs), boundary conditions and interface conditions. In Section 3, we give a brief description of our discretization approach and describe the coupled interface (CI) and segregated interface (SI) approaches. Our approach for multi-domain time-stepping is presented in Section 4. Section 5 provides a stability analysis of the coupled interface approximation for a model problem. In Section 6, we analyze the SI technique using Dirichlet–Neumann interface conditions and derive the convergence characteristics of the iteration. We also analyze a mixed interface approximation and present computed convergence rates for various cases. In Section 7, we present numerical results that show the accuracy of the method and its use applied to some interesting applications. Some parallel scaling results are also provided. Conclusions are given in the final section.

The computations in this paper were performed with the composite grid multi-physics solver *cgmp* together with the incompressible Navier–Stokes solver *cgins* and the advection–diffusion (and heat equation) solver *cgad*. These programs are part of the *CG* suite of PDE solvers and are built upon the *Overture* framework. These programs are all freely available at http://www.llnl.gov/casc/Overture.

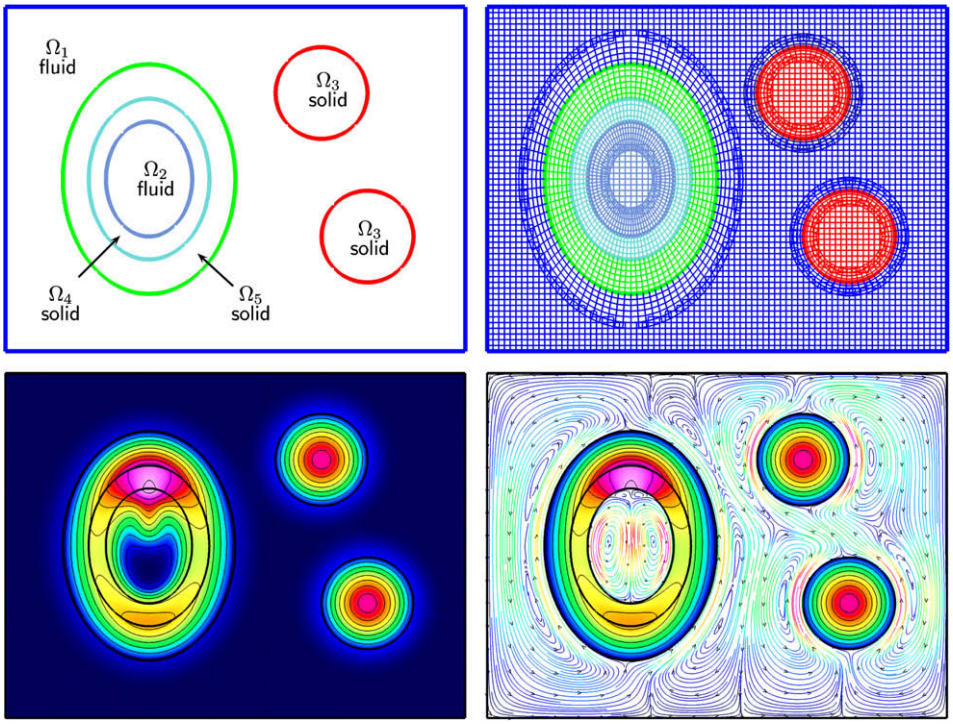

Fig. 1. Top left : a domain Ω with two fluid sub-domains Ω_1, Ω_2 and three solid sub-domains Ω_3, Ω_4 and Ω_5. Top right: the composite $\mathcal{G}$ consists of five overlapping grids, one for each domain. Bottom left: a computed conjugate heat transfer solution showing the temperature. Bottom right: the streamlines in the fluid domains and the temperature in the solid domains.

2. Problem specification and model equations

We are interested in solving a conjugate heat transfer problem in a domain Ω which consists of a set of $\mathcal{N}_d$ sub-domains that represent fluid and solid regions (see Fig. 1),

$$\Omega = \cup_{d=1}^{\mathcal{N}_d} \Omega_d. \tag{1}$$

The sub-domains are assumed to be non-overlapping. A single sub-domain, however, may be multiply connected. A collection of non-overlapping regions may be represented, for example, with a single sub-domain (as Ω_3 in Fig. 1). We define the sub-domains in this way since we associate a separate physics solver with each sub-domain. A single heat equation solver, for example, could be used on a multiply connected domain. This is described further in Section 3.

The solution in a fluid domain, Ω_f, with boundary $\partial\Omega_f$, is governed by the incompressible Navier–Stokes (INS) equations. The effects of temperature and buoyancy are modeled with the Boussinesq approximation. The equations are given by

$$\begin{cases} \mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \nu\Delta\mathbf{u} + \alpha\mathbf{g}(T - T_{\text{ref}}) - \mathbf{f} = 0, & t > 0,\ \mathbf{x}\in\Omega_f, \\ \Delta p + \nabla\mathbf{u} : \nabla\mathbf{u} + \alpha(\mathbf{g}\cdot\nabla)T - \nabla\cdot\mathbf{f} = 0, & t > 0,\ \mathbf{x}\in\Omega_f, \\ T_t + (\mathbf{u}\cdot\nabla)T - \dfrac{1}{\rho C}\nabla\cdot(\mathcal{K}\nabla T) - f_T = 0, & t > 0,\ \mathbf{x}\in\Omega_f \end{cases} \tag{2}$$

with initial conditions and boundary conditions,

$$\begin{cases} (\mathbf{u}(\mathbf{x},0), T(\mathbf{x},0)) = (\mathbf{u}_I(\mathbf{x}), T_I(\mathbf{x})), & t = 0,\ \mathbf{x}\in\Omega_f, \\ \mathcal{B}^F(\mathbf{u}, T) = 0, & t > 0,\ \mathbf{x}\in\partial\Omega_f. \end{cases} \tag{3}$$

Here $u = u(\mathbf{x}, t)$ is the velocity, p the kinematic pressure, T the temperature, $\nu = \mu/\rho$ the kinematic viscosity, $\mathcal{K}$ the thermal conductivity, ρ the density, C the specific heat, α the coefficient of thermal expansion and T_{ref} is the reference temperature for which there are no buoyancy forces. $\mathbf{f}$ and f_T are forcing functions. We use a pressure–velocity formulation for these equations, solved with a split-step method. A second-order accurate and fourth-order accurate scheme are available, see [44–46] for more details. Boundary conditions for these equations are discussed in Section 3.1. We have also developed efficient multigrid algorithms for overlapping grids [47] that can be used with the incompressible flow solver, although the multigrid solver is not used in the present work.

Heat conduction in a solid domain Ω_s is modeled by the heat equation,

$$\begin{cases} T_t - \dfrac{1}{\rho C}\nabla\cdot(\mathcal{K}\nabla T) - f = 0, & t > 0,\ \mathbf{x}\in\Omega_s, \\ T(\mathbf{x},0) = T_I(\mathbf{x}), & t = 0,\ \mathbf{x}\in\Omega_s, \\ \mathcal{B}^S(T) = 0, & t > 0,\ \mathbf{x}\in\partial\Omega_s. \end{cases} \tag{4}$$

At the interface $\mathcal{I} = \partial\Omega_f \cap \partial\Omega_s$ between a fluid region and a solid region the matching conditions are the continuity of the temperature and the continuity of the normal component of the heat flux

$$[T]_{\mathcal{I}} = 0, \tag{5}$$

$$[\mathcal{K}\mathbf{n}\cdot\nabla T]_{\mathcal{I}} = 0. \tag{6}$$

Here $\mathbf{n}$ is the normal to the interface $\mathcal{I}$ and $[\cdot]_{\mathcal{I}}$ denotes the jump across the interface. The interface conditions (5) and (6) also apply at the interface between two fluid regions or between two solid regions. Higher-order accurate methods may require higher-order matching conditions. These are derived by taking time derivatives of basic jump conditions (5) and (6) and using the governing equation to replace time derivatives with space derivatives. Assuming for simplicity that the coefficients $\mathcal{K}$, ρ and C are constant within a sub-domain, the high-order jump conditions are

$$\left[(\mathcal{D}\Delta)^q T + \sum_{j=0}^{q-1}(\mathcal{D}\Delta)^{q-1-j}\partial_t^j f\right]_{\mathcal{I}} = 0, \quad q = 0,1,2,\ldots, \tag{7}$$

$$\left[\mathcal{K}(\mathcal{D}\Delta)^q\mathbf{n}\cdot\nabla T + \mathcal{K}\sum_{j=0}^{q-1-k}(\mathcal{D}\Delta)^{q-1}\partial_t^j\mathbf{n}\cdot\nabla f\right]_{\mathcal{I}} = 0, \quad q = 0,1,2,\ldots, \tag{8}$$

where the thermal diffusivity $\mathcal{D}$ is defined as

$$\mathcal{D} = \mathcal{K}/(\rho C). \tag{9}$$

In this paper we only present results for second-order accurate approximations although we will use condition (7) with $q = 1$ for the coupled interface approach described in Section 3.2.

3. Solution approach and discretization

We solve the multi-domain problem using a weakly coupled approach (although when explicit time-stepping is used the approach is effectively strongly coupled). We associate a separate physics solver with each fluid or solid sub-domain of the

multi-domain problem. We use the composite grid incompressible Navier–Stokes code *cgins* in a fluid domain and the advection–diffusion code *cgad* in a solid domain. There may be multiple instances of *cgins* and *cgad*. The composite grid multi-physics solver *cgmp* manages the multi-domain solution process and coordinates the transfer of information at the interfaces. The multi-domain time-stepping algorithm is discussed in more detail in Section 4.

The entire domain of interest, Ω, is discretized using a composite overlapping grid, $\mathcal{G}$. Each fluid or solid sub-domain will itself be discretized with an overlapping grid, $\mathcal{G}_d$, with the global overlapping grid $\mathcal{G}$ containing all the sub-overlapping grids,

$$\mathcal{G} = \cup_{d=1}^{\mathcal{N}_d} \mathcal{G}_d. \tag{10}$$

The overlapping grid $\mathcal{G}$ consists of a set of $\mathcal{N}_{\text{grid}}$ component grids G_g, i.e.,

$$\mathcal{G} = \{G_g\}, \quad g = 1, 2, \ldots, \mathcal{N}_{\text{grid}}.$$

The component grids cover Ω. Similarly, the overlapping grid, $\mathcal{G}_d$, for a sub-domain, will consist of a set of overlapping component grids which are a sub-set of the grids in $\mathcal{G}$.

Each component grid is a logically rectangular, curvilinear grid defined by a smooth mapping $\mathbf{C}_g$ from parameter space $\mathbf{r}$ (e.g. the unit-cube in three-dimensions) to physical space $\mathbf{x}$:

$$\mathbf{x} = \mathbf{C}_g(\mathbf{r}), \quad \mathbf{r} \in [0,1]^3, \ \mathbf{x} \in \mathbb{R}^3.$$

The mapping is used to define the metric derivatives $\partial\mathbf{x}/\partial\mathbf{r}$ and the grid points at any desired resolution. Variables defined on a component grid, such as the coordinates of the grid points, are stored in rectangular arrays. For example, grid vertices are represented as the array

$$\mathbf{x}^g_{\mathbf{i}}: \ \text{grid vertices}, \quad \mathbf{i} = (i_1, i_2, i_3), \ i_k = 0, \ldots, N_k, \ k = 1, 2, 3,$$

where N_k is the number of grid cells in k-coordinate direction. We note that grid vertex information and other mapping information are not stored for Cartesian grids. This usually results in a considerable savings in memory use since most of the grid points belong to Cartesian grids for a typical overlapping grid.

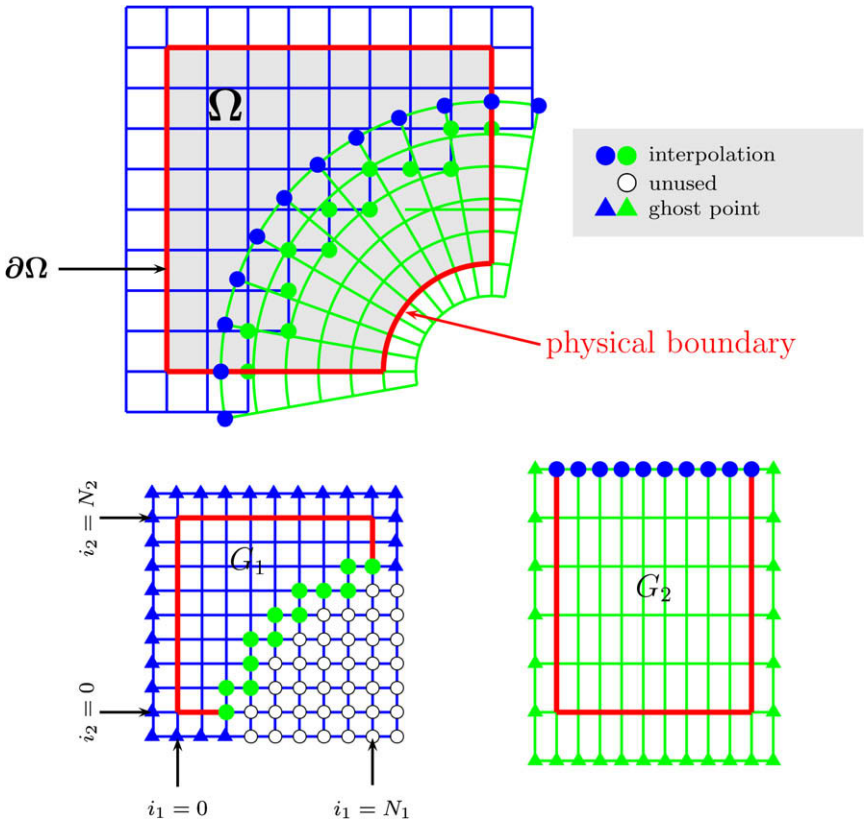

Fig. 2. The top view shows an overlapping grid consisting of two structured curvilinear component grids. The bottom views show the component grids in the unit square parameter space. Grid points are classified as discretization points, interpolation points or unused points. Ghost points are used to apply boundary conditions.

Fig. 2 shows a simple overlapping grid consisting of two component grids, an annular boundary fitted grid and a background Cartesian grid. The top view shows the overlapping grid while the bottom view shows each grid in parameter space. In this example the annular grid cuts a hole in the Cartesian grid so that the latter grid has a number of unused points which are marked as open circles. The other points on the component grids are classified as either discretization points (where the PDE or boundary conditions are discretized) or interpolation points. This information is supplied by the overlapping grid generator Ogen [48] and is held in an integer mask array. In addition, each boundary face of each component grid is classified as either a physical boundary (where boundary conditions are to be implemented), a periodic boundary or an interpolation boundary. Typically, one or more layers of ghost points are created for each component grid to aid in the application of boundary conditions.

The PDE's that define the evolution of the fluid and solid are discretized with a finite difference or finite-volume approximation with all variables defined at the nodes. For example, consider approximating the generalized Laplace operator, L defined by

$$Lw = \nabla \cdot (a \nabla w), \tag{11}$$

where $a = a(\mathbf{x})$ is a real valued coefficient. A straight-forward approach to discretize L on a curvilinear grid is to use the mapping method, as follows. Using the chain rule, the operator L can be written in general curvilinear coordinates in n_d space-dimensions as

$$Lw = \sum_{i=1}^{n_d}\sum_{j=1}^{n_d}\sum_{k=1}^{n_d} a \frac{\partial r_k}{\partial x_i}\frac{\partial r_j}{\partial x_i}\frac{\partial^2 w}{\partial r_j \partial r_k} + \frac{\partial r_k}{\partial x_i}\left\{ a \frac{\partial}{\partial r_k}\left(\frac{\partial r_j}{\partial x_i}\right) + \frac{\partial a}{\partial r_k}\frac{\partial r_j}{\partial x_i}\right\}\frac{\partial w}{\partial r_j}. \tag{12}$$

The metric terms $\partial r_j/\partial x_k$ are computed from the mapping that defines the grid and are thus assumed to be known. The derivatives with respect to the parameter space coordinates r_j can be approximated with central difference approximations. The operator L can also be written in *conservation* form, or *self-adjoint* form, in general curvilinear coordinates as

$$Lw = \frac{1}{J}\sum_{j=1}^{n_d}\sum_{k=1}^{n_d}\frac{\partial}{\partial r_j}\left(A^{jk}\frac{\partial w}{\partial r_k}\right), \quad A^{jk} = a\, J \sum_{\mu=1}^{n_d}\sum_{\nu=1}^{n_d}\frac{\partial r_j}{\partial x_\mu}\frac{\partial r_k}{\partial x_\nu}, \tag{13}$$

where J denotes the determinant of the Jacobian matrix $[\partial x_j/\partial r_k]$. A careful discretization of this last form of the operator leads to symmetric and compact discrete approximations of any order of accuracy [26]. These approximations are generalized finite-volume approximations. See [44] for more details on discretizing the incompressible Navier–Stokes equations on curvilinear grids.

3.1. Boundary conditions

The boundary conditions we use for the incompressible Navier–Stokes equations and the heat equation are given in Figs. 3 and 4, respectively. We consider inflow boundaries, outflow boundaries, no-slip walls and slip-walls. The conditions imposed on each boundary are divided into those labeled *physical* which come from the analytic definition of a well-posed initial-boundary-value problem and those labeled *numerical* which are extra conditions needed for the discretized problem in order to define an accurate and stable approximation. Numerical boundary conditions typically determine ghost point values of the discrete solution. Note that there is no explicit *physical* boundary condition for the pressure at walls or inflow. The boundary condition $\nabla \cdot \mathbf{u} = 0$, which takes the place of an explicit condition on the pressure, ensures that the solution to the INS Eqs. (2) and (3) satisfies $\nabla \cdot \mathbf{u} = 0$ everywhere. The *numerical* boundary condition for the pressure at walls or inflow is

$$p_n = \mathcal{P}_b(\mathbf{u}, T) \equiv -\mathbf{n} \cdot (\nu \nabla \times \nabla \times \mathbf{u} + \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \alpha \mathbf{g} T), \tag{14}$$

and is derived from the normal component of the momentum equations. This numerical boundary condition is used when solving the Poisson equation for the pressure. See [49,46] for a discussion of this boundary condition. The boundary condi-

INS	no-slip wall or inflow
physical	$\mathbf{u} = \mathbf{g}_{\mathbf{u}}$
	$\nabla \cdot \mathbf{u} = 0$
	$a\,T + b\,T_n = g_T$
numerical	$p_n = \mathcal{P}_b(\mathbf{u}, T)$
	extrapolate $\boldsymbol{\tau}_\mu \cdot \mathbf{u}$
	extrapolate T if $b = 0$

INS	slip wall
physical	$\mathbf{n} \cdot \mathbf{u} = g_n$
	$\nabla \cdot \mathbf{u} = 0$
	$a\,T + b\,T_n = g_T$
numerical	$p_n = \mathcal{P}_b(\mathbf{u}, T)$
	extrapolate $\boldsymbol{\tau}_\mu \cdot \mathbf{u}$
	extrapolate T if $b = 0$

INS	outflow
physical	$c\,p_n + d\,p = g_p$
numerical	extrapolate $\mathbf{u}$
	extrapolate T

Fig. 3. Boundary conditions for the incompressible Navier–Stokes equations. The outward normal at the boundary is denoted by $\mathbf{n}$, tangent vectors at the boundary are τ_μ, $\mu = 1, 2$ and $\mathcal{P}_b$ is defined by Eq. (14). The functions $\mathbf{g}_{\mathbf{u}}(\mathbf{x}, t), \mathbf{g}_n(\mathbf{x}, t), g_T(\mathbf{x}, t)$ and $g_p(\mathbf{x}, t)$ are given forcing functions while a,b, c and d are non-negative constants with $a + b > 0$ and $c + d > 0$.

Heat Equation	wall
physical	$a\,T + b\,T_n = g_T$
numerical	extrapolate T if $b = 0$

Fig. 4. Boundary conditions for the heat equation, where $g_T(\mathbf{x}, t)$ is a given forcing function and a and b are non-negative constants with $a + b > 0$.

tion on the temperature can be Dirichlet, $T = g_T$, Neumann, $T_n = g_T$, or a mixed condition $aT_n + bT = g_T$. For further details on the discretization of the boundary conditions, the reader is referred to Henshaw [44], Henshaw and Petersson [46] and Henshaw et al. [45].

3.2. Explicit time-stepping and coupled interface equations (CI)

We consider two approaches for solving the interface equations. In the first approach the interface equations are solved as a coupled system of equations using a *centered approximation*. This approach is generally used when the interior equations are advanced with explicit time-stepping and requires no iteration on the interface values. The CI approach could be used directly with implicit time-stepping but would require the formation of a coupled implicit system of equations for the temperature across all domains. For a second-order accurate approximation we use the two interface equations (6) and (7) with $q = 1$,

$$\begin{aligned} &[\mathcal{K}\mathbf{n}\cdot\nabla T]_{\mathcal{I}} = 0, \\ &[\mathcal{D}\Delta T + f]_{\mathcal{I}} = 0. \end{aligned} \tag{15}$$

These two conditions will determine the discrete solution values on the first ghost line of the two domains, one ghost line for each domain. Here we have assumed that $\mathbf{u} = 0$ on the interface and that the coefficients $\mathcal{K}$, ρ and C are constant within each sub-domain. This approximation differs from those usually found in the literature through the use of Eq. (15). For the case of an interface between two solid regions, for example, the time-continuous space-discrete approximation is given by

$$\partial_t T_{m,\mathbf{i}} = \mathcal{D}_m \Delta_h T_{m,\mathbf{i}} + f_m(\mathbf{x}_{m,\mathbf{i}}, t), \qquad \text{for } \mathbf{i} \in G_m^d, \text{ and sub-domains } m = 1,2, \tag{16}$$

$$\mathcal{K}_1 \mathbf{n}_{1,\mathbf{i}} \cdot \nabla_h T_{1,\mathbf{i}} = \mathcal{K}_2(-\mathbf{n}_{2,\mathbf{j}}) \cdot \nabla_h T_{2,\mathbf{j}}, \qquad \text{for } \mathbf{i} \in \mathcal{I}_h^{1,2},\ \mathbf{j} \in \mathcal{I}_h^{2,1}, \tag{17}$$

$$\mathcal{D}_1 \Delta_h T_{1,\mathbf{i}} + f_1(\mathbf{x}_{\mathbf{i}}, t) = \mathcal{D}_2 \Delta_h T_{2,\mathbf{j}} + f_2(\mathbf{x}_{2,j_v}, t), \quad \text{for } \mathbf{i} \in \mathcal{I}_h^{1,2},\ \mathbf{j} \in \mathcal{I}_h^{2,1}. \tag{18}$$

Here $T_{m,\mathbf{i}} \approx T(\mathbf{x}_{m,\mathbf{i}}, t)$ denotes the discrete approximation to the temperature on a grid G_m of sub-domain Ω_m with grid points $\mathbf{x}_{m,\mathbf{i}}$ and interface normals $\mathbf{n}_{m,\mathbf{i}}$. Δ_h and ∇_h denote discrete approximations to Δ and ∇, respectively. These operators depend on the grid G_m^d they are discretized on, but this should be clear from context. Note that we use $-\mathbf{n}_{2,\mathbf{j}}$ since the discrete normals are defined as outward normals. Equations are also needed at physical boundaries, interpolation points and periodic boundaries but these are left out in order to focus on the interface treatment. The interior equation (16) is applied at interior discretization points, boundary points and interface points, the set of these grid points being denoted by G_m^d. The set of points on component grid G_p that lie on the interface with grid G_q are denoted by $\mathcal{I}_h^{p,q}$. We have assumed that the grid points align on the interface so that the sets $\mathcal{I}_h^{p,q}$ and $\mathcal{I}_h^{q,p}$ define the same set of physical points. This assumption is used throughout this paper. The more general case of non-matching grid points on the interface is left to future work. The interface equations (17) and (18) are discrete approximations that are centered on the interface and will be used to determine the values on the ghost points that are adjacent to the interface points. Use of centered schemes for boundaries generally results in more accurate and more stable approximations compared to using one-sided approximations based on extrapolation.

We now illustrate how the discrete solution is advanced in time using the forward-Euler method. The discrete solution, $T_{m,\mathbf{i}}^n \approx T(\mathbf{x}_{m,\mathbf{i}}, t^n)$, with $t^n = n\Delta t$, is first advanced at all interior and interface points using Eq. (16),

$$T_{m,\mathbf{i}}^{n+1} = T_{m,\mathbf{i}}^n + \Delta t(\mathcal{D}_m \Delta_h T_{m,\mathbf{i}}^n + f_m(\mathbf{x}_{m,\mathbf{i}}, t^n)), \quad \text{for } \mathbf{i} \in G_m^d,\ m = 1,2. \tag{19}$$

Eqs. (17) and (18) are then used to determine the ghost point values adjacent to the interface at the new time level,

$$\mathcal{K}_1 \mathbf{n}_{1,\mathbf{i}} \cdot \nabla_h T_{1,\mathbf{i}}^{n+1} = \mathcal{K}_2(-\mathbf{n}_{2,\mathbf{j}}) \cdot \nabla_h T_{2,\mathbf{j}}^{n+1}, \qquad \text{for } \mathbf{i} \in \mathcal{I}_h^{1,2},\ \mathbf{j} \in \mathcal{I}_h^{2,1}, \tag{20}$$

$$\mathcal{D}_1 \Delta_h T_{1,\mathbf{i}}^{n+1} + f_1(\mathbf{x}_{\mathbf{i}}, t^{n+1}) = \mathcal{D}_2 \Delta_h T_{2,\mathbf{j}}^{n+1} + f_2(\mathbf{x}_{2,\mathbf{j}}, t^{n+1}), \quad \text{for } \mathbf{i} \in \mathcal{I}_h^{1,2},\ \mathbf{j} \in \mathcal{I}_h^{2,1}. \tag{21}$$

In practice we often use a predictor–corrector time-stepping method in which case the interface equations are applied after the predictor and corrector steps. Application of Eqs. (19)–(21) should ensure that $T_{1,\mathbf{i}}^{n+1} = T_{2,\mathbf{j}}^{n+1}$ (to round-off error) for points on the interface. On non-orthogonal grids, condition (21) will couple the solutions on the ghost points due to cross-derivative terms in the approximation to Δ_h. To avoid solving a coupled system of equations along the interface we use the current best guess values for ghost points values when they are needed by the cross-derivatives (using extrapolation in space for the

predictor step). We then enforce the continuity of temperature directly after each time-step using a weighted average of the computed interface values, following the approximation developed by Patankar [33],

$$T^{n+1}_{1,\mathbf{i}} = T^{n+1}_{2,\mathbf{j}} = \frac{\mathcal{K}_1 \widetilde{T}^{n+1}_{1,\mathbf{i}} + \mathcal{K}_2 \widetilde{T}^{n+1}_{2,\mathbf{j}}}{\mathcal{K}_1 + \mathcal{K}_2}, \quad \text{for } \mathbf{i} \in \mathcal{I}^{1,2}_h,\ \mathbf{j} \in \mathcal{I}^{2,1}_h, \tag{22}$$

where $\widetilde{T}^{n+1}_{1,\mathbf{i}}$ and $\widetilde{T}^{n+1}_{2,\mathbf{j}}$ are the values determined from Eqs. (19)–(21). The stability and accuracy of the approximations (16)–(18) are studied in Section 5. Unlike the segregated approach, discussed in the next section, the stability of the coupled approach does not depend on the relative sizes of $\mathcal{K}$ and $\mathcal{D}$ in the two domains.

3.3. Implicit time-stepping and segregated interface equations (SI)

The second approach used to solve the interface equations is based on segregating the interface equations and applying one of the interface equations as a boundary condition for one domain and the other as the boundary condition for the second domain. We generally use this approach when one or both of the domains uses implicit time-stepping. A segregated approach is commonly used in the literature with both explicit and implicit time-stepping. Some care is required in applying the segregated method since the stability of the method depends on which interface equation is associated with which domain and on the relative sizes of $\mathcal{K}$ and $\mathcal{D}$ in the two domains as discussed in Section 6.

In the segregated approach the solution on each sub-domain is advanced using one of the interface conditions as a boundary condition. For example, the solution on one domain may be advanced for one time-step using an implicit method with a Dirichlet condition that sets the temperature on the interface

$$T^{n+1}_{1,\mathbf{i}} = \mathcal{L}^1(T^{n+1}_{1,\mathbf{i}}, T^n_{1,\mathbf{i}}, \ldots), \quad \text{for } \mathbf{i} \in G^d_1, \tag{23}$$

$$T^{n+1}_{1,\mathbf{i}} = T^*_{2,\mathbf{j}}, \qquad \text{for } \mathbf{i} \in \mathcal{I}^{1,2}_h,\ \mathbf{j} \in \mathcal{I}^{2,1}_h. \tag{24}$$

Here $T^*_{2,\mathbf{j}}$ is some guess at the temperature on the interface and $\mathcal{L}^1$ denotes an implicit time-stepping operator. Note that we apply the interior equation on the interface in addition to the Dirichlet condition and this extra equation determines the solution on the ghost points:

$$\mathcal{L}^1(T^{n+1}_{1,\mathbf{i}}, T^n_{1,\mathbf{i}}, \ldots) = T^*_{2,\mathbf{j}}, \quad \text{for } \mathbf{i} \in \mathcal{I}^{1,2}_h,\ \mathbf{j} \in \mathcal{I}^{2,1}_h. \tag{25}$$

The solution on the adjacent domain is advanced using a Neumann boundary condition based on the continuity of heat flux as an interface condition

$$T^{n+1}_{2,\mathbf{i}} = \mathcal{L}^2(T^{n+1}_{2,\mathbf{i}}, T^n_{2,\mathbf{i}}, \ldots), \qquad \text{for } \mathbf{i} \in G^d_2, \tag{26}$$

$$\mathcal{K}_2(-\mathbf{n}_{2,\mathbf{j}}) \cdot \nabla_h T^{n+1}_{2,\mathbf{j}} = \mathcal{K}_1 \mathbf{n}_{1,\mathbf{i}} \cdot \nabla_h T^*_{1,\mathbf{i}}, \quad \text{for } \mathbf{i} \in \mathcal{I}^{1,2}_h,\ \mathbf{j} \in \mathcal{I}^{2,1}_h, \tag{27}$$

where some guess for the heat flux from domain one is used. In general we will iterate these equations some number of times using successively better values for $\mathbf{n}_{1,\mathbf{i}} \cdot \nabla_h T^*_{1,\mathbf{i}}$ and $T^*_{2,\mathbf{j}}$. If we iterate to convergence, the results will satisfy the same centered interface conditions (20) and (21) that are satisfied in the coupled approach.

In Section 6, we analyze this Dirichlet–Neumann segregated approach and discuss iteration strategies. We also consider a generalization that uses a mixed (Robin) approximation on both sides of the interface.

4. The multi-domain time-stepping algorithm

In this section we describe our approach for time-stepping a multi-domain problem. We assume that we have a separate physics solver for each domain. For the purposes of this paper the domain solver will either solve the INS equations or the heat equation. There may be multiple INS solvers and multiple heat equation solvers. The different domain solvers are coupled through the interface conditions (5) and (6). These interface equations are solved with a coupled (CI) or segregated (SI) approach as described in Sections 3.2 and 3.3, respectively. Each domain solver is assumed to time-step its equations with an explicit or implicit time-stepping technique such as forward-Euler, backward-Euler, or an (implicit/explicit) predictor–corrector. To be more concrete we suppose that each domain solver uses a time-stepping approach that is of the form of the following generic implicit predictor–corrector algorithm given by

$$\begin{aligned}
&L_p \mathbf{v}^{(0)} = \mathbf{f}_p(\mathbf{u}^n, \mathbf{u}^{n-1}, \ldots) \quad \text{(predictor)},\\
&L_c \mathbf{v}^{(k)} = \mathbf{f}_c(\mathbf{v}^{(k-1)}, \mathbf{u}^n, \ldots) \quad \text{(corrector, } k = 1, 2, \ldots, n_c),\\
&\mathbf{u}^{n+1} = \mathbf{v}^{(n_c)}.
\end{aligned}$$

Here $\mathbf{u}^n$ is the solution at time t^n, $\mathbf{v}^{(k)}$ are intermediate solution values, and L_p and L_c denote possibly implicit operators that are solved at the predictor and corrector steps, respectively. When we solve the interface equations by iteration, we may need to take n_i additional corrector steps over and above the number, n_c, used by default in the above scheme.

A one-dimensional model problem on the interval $(-\infty,\infty)$ with material interface at $x=0$ is

$$\partial_t u_1 = \mathcal{D}_1\partial_x^2 u_1, \qquad \text{for } x<0, \tag{28}$$
$$\partial_t u_2 = \mathcal{D}_2\partial_x^2 u_2, \qquad \text{for } x>0, \tag{29}$$
$$u_1 - u_2 = 0, \qquad \text{at } x=0,, \tag{30}$$
$$\mathcal{K}_1\partial_x u_1 - \mathcal{K}_2\partial_x u_2 = 0, \qquad \text{at } x=0, \tag{31}$$
$$u_1(x,0) = u_1^0(x), \quad u_2(x,0) = u_2^0(x), \quad \text{at } t=0, \tag{32}$$
$$\|u_1\|<\infty, \quad \|u_2\|<\infty, \tag{33}$$

where $u_1 = u_1(x,t)$ is defined on the left semi-infinite interval $(\infty,0]$ and $u_2 = u_2(x,t)$ on the right semi-infinite interval $[0,\infty)$. The initial conditions are assumed to have compact support and $\|u_m\|$, $m=1,2$, denotes the L_2 norm of u_m over the appropriate interval. As discussed previously, by taking the time derivative of the jump condition (30) and using the interior equation it follows that

$$\mathcal{D}_1\partial_x^2 u_1 - \mathcal{D}_2\partial_x^2 u_2 = 0, \quad \text{at } x=0. \tag{34}$$

We assume that the initial conditions satisfy consistency conditions such as $u_1^0(0) = u_2^0(0)$ and $\mathcal{D}_1\partial_x^2 u_1^0(0) = \mathcal{D}_2\partial_x^2 u_2^0(0)$.

Introduce a one-dimensional overlapping grid as shown in Fig. 6. The domain is discretized with a grid with spacing h_1 for $x<0$ and a grid with spacing h_2 for $x>0$. Let $x_{1,j} = jh_1$ and $x_{2,j} = jh_2$ denote the grid points on either side of the interface. A (formally) second-order accurate semi-discrete approximation is

$$\partial_t U_{1,j} = \mathcal{D}_1 D_+ D_- U_{1,j}, \quad \text{for } j=0,-1,-2\ldots, \tag{35}$$
$$\partial_t U_{2,j} = \mathcal{D}_2 D_+ D_- U_{2,j}, \quad \text{for } j=0,1,2,\ldots, \tag{36}$$
$$\mathcal{K}_1 D_0 U_{1,0} = \mathcal{K}_2 D_0 U_{2,0}, \tag{37}$$
$$\mathcal{D}_1 D_+ D_- U_{1,0} = \mathcal{D}_2 D_+ D_- U_{2,0}, \tag{38}$$
$$U_{1,j}(0) = u_1^0(x_{1,j}), \quad \text{for } j=1,0,-1,-2\ldots, \tag{39}$$
$$U_{2,j}(0) = u_2^0(x_{2,j}), \quad \text{for } j=-1,0,1,2,\ldots, \tag{40}$$
$$\|U_1\|_h<\infty, \quad \|U_2\|_h<\infty. \tag{41}$$

Here $U_{1,j}(t)$ and $U_{2,j}(t)$ denote the approximations to u_1 and u_2 on the grids while $D_+U_{m,j} = (U_{m,j+1} - U_{m,j})/h_m$, $D_-U_{m,j} = (U_{m,j} - U_{m,j-1})/h_m$ and $D_0U_{m,j} = (U_{m,j+1} - U_{m,j-1})/(2h_m)$ denote the usual finite difference operators [50]. The discrete norms of the grid functions are defined from $\|U_1\|_h^2 = \sum_{j=-\infty}^{0}|U_{1,j}|^2h_1$ and $\|U_2\|_h^2 = \sum_{j=0}^{\infty}|U_{2,j}|^2h_2$. Approximations to the jump conditions (31) and (34) have been imposed. The basic jump condition $U_{1,0}(t) = U_{2,0}(t)$ will also hold in the discrete case since $\partial_t U_{1,0} = \partial_t U_{2,0}$ for all time by (35) and (38), and since we assume the initial conditions satisfy (38) and $U_{1,0}(0) = U_{2,0}(0)$.

Theorem 1. *The solution to the one-dimensional interface problem (35)–(41) is stable and second-order accurate.*

Proof. See Appendix A. □

Note: The interface equations (37) and (38) can be solved for the values at the ghost points, $U_{1,+1}$, $U_{2,-1}$ in terms of interior values,

$$U_{1,+1} = [2(\widetilde{\mathcal{D}}_1 - \widetilde{\mathcal{D}}_2)\widetilde{\mathcal{K}}_2U_{2,0} + 2\widetilde{\mathcal{D}}_2\widetilde{\mathcal{K}}_2U_{2,1} + (\widetilde{\mathcal{D}}_2\widetilde{\mathcal{K}}_1 - \mathcal{D}_1\widetilde{\mathcal{K}}_2)U_{1,-1}]/[\widetilde{\mathcal{D}}_1\widetilde{\mathcal{K}}_2 + \widetilde{\mathcal{D}}_2\widetilde{\mathcal{K}}_1], \tag{42}$$
$$U_{2,-1} = [2(\widetilde{\mathcal{D}}_2 - \widetilde{\mathcal{D}}_1)\widetilde{\mathcal{K}}_1U_{1,0} + 2\widetilde{\mathcal{D}}_1\widetilde{\mathcal{K}}_1U_{1,-1} + (\widetilde{\mathcal{D}}_1\widetilde{\mathcal{K}}_2 - \mathcal{D}_2\widetilde{\mathcal{K}}_1)U_{2,1}]/[\widetilde{\mathcal{D}}_1\widetilde{\mathcal{K}}_2 + \widetilde{\mathcal{D}}_2\widetilde{\mathcal{K}}_1], \tag{43}$$

where $\widetilde{\mathcal{D}}_m = \mathcal{D}_m/h_m^2$ and $\widetilde{\mathcal{K}}_m = \mathcal{K}_m/h_m$. The interface boundary conditions thus have the useful property that when $\mathcal{K}_1 = \mathcal{K}_2$, $\mathcal{D}_1 = \mathcal{D}_2$ and $h_1 = h_2$ the discrete solution is the same as if there were no interface at all: $U_{1,1} = U_{2,1}$ and $U_{2,-1} = U_{1,-1}$.

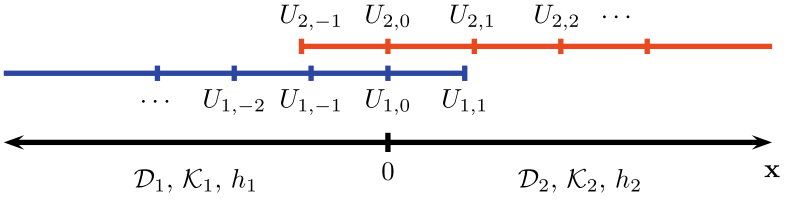

Fig. 6. The overlapping grids for a one-dimensional material interface located at $x=0$. The ghost points $U_{2,-1}$ and $U_{1,1}$ are introduced on the two grids at the interface.

We have shown that the time-continuous and space-discrete problem is stable with the centered interface conditions. We now discretize in time with forward-Euler and show that for the fully discrete case the scheme is stable provided the time-step Δt is chosen to satisfy the usual von Neumann stability conditions for each side of the interface. Discretizing Eqs. (35)–(41) using forward-Euler in time gives the approximation

$$(U^{n+1}_{1,j} - U^n_{1,j})/\Delta t = \mathcal{D}_1 D_+ D_- U^n_{1,j}, \quad \text{for } j = 0,-1,-2,\ldots, \tag{44}$$

$$(U^{n+1}_{2,j} - U^n_{2,j})/\Delta t = \mathcal{D}_2 D_+ D_- U^n_{2,j}, \quad \text{for } j = 0,1,2,\ldots, \tag{45}$$

$$\mathcal{K}_1 D_0 U^{n+1}_{1,0} = \mathcal{K}_2 D_0 U^{n+1}_{2,0}, \tag{46}$$

$$\mathcal{D}_1 D_+ D_- U^{n+1}_{1,0} = \mathcal{D}_2 D_+ D_- U^{n+1}_{2,0}, \tag{47}$$

$$U^0_{1,j} = u^0_1(x_{1,j}), \quad \text{for } j = 1,0,-1,-2,\ldots, \tag{48}$$

$$U^0_{2,j} = u^0_2(x_{2,j}), \quad \text{for } j = -1,0,1,2,\ldots, \tag{49}$$

$$\|U^{n+1}_1\|_h < \infty, \quad \|U^{n+1}_2\|_h < \infty, \tag{50}$$

where $U^n_{m,j} \approx u_m(x_{m,j}, t^n)$. These equations will be stable in the sense of Godunov–Ryabenkii [50] provided there are no solutions to the homogeneous equations (i.e. with $u^0_1(x_{1,j}) = 0$ and $u^0_2(x_{2,j}) = 0$) of the form

$$U^n_{1,j} = A z^n \hat{U}_{1,j}, \tag{51}$$

$$U^n_{2,j} = B z^n \hat{U}_{2,j} \tag{52}$$

with $|z| > 1$.

Theorem 2. *Solutions to Eqs. (44)–(50) are stable in the sense of Godunov–Ryabenkii provided the time-step Δt satisfies the von Neumann stability conditions*

$$d_m \equiv \frac{\mathcal{D}_m \Delta t}{h^2_m} < \frac{1}{2}, \quad m = 1,2. \tag{53}$$

Proof. See Appendix A. □

The results of Theorems 1 and 2 are shown to apply to more complex two- and three-dimensional conjugate heat transfer problems in Section 7. These results show that the coupled interface approach with second-order explicit time-stepping is second-order accurate in space and time. Furthermore, we note that in all the cases considered, the coupled method is stable provided the time-step is chosen as the minimum of the time steps required to make the sub-domain solvers stable. Thus there is no need to reduce the time-step due to any effects of the interface approximation.

6. Domain implicit time-stepping and the solution of the interface equations by iteration

In this section we analyze the segregated interface method, as described in Section 3.3, for solving the interface equations when the sub-domains are advanced in a weakly coupled fashion with implicit time-stepping. We suppose that we can advance each sub-domain problem separately (with some appropriate conditions at the interface) but we do not want to solve the full coupled implicit problem. Implicit methods require interface values at the new time level and some approximate values must be provided. We define an iteration, as part of a predictor–corrector scheme, to solve the coupled problem. In Section 6.1 we study the commonly used Dirichlet–Neumann (DN) approach that uses a Dirichlet condition at the interface of one domain and a Neumann condition at the interface of the second domain. The convergence properties of this iteration are analyzed. In Section 6.2 we analyze the case when a mixed condition is applied at both interfaces and show that this approach has some advantages over the DN approach.

6.1. Analysis of the segregated interface (SI) method

We consider the model problem of solving the heat equation in two domains separated by an interface using an implicit time-stepping method. The heat conduction model problem is defined on the region Ω consisting of two adjacent squares, $\Omega = (-a,b)\times(0,2\pi) = \Omega_1 \bigcup \Omega_2$ where the left domain is $\Omega_1 = (-a,0)\times(0,2\pi)$, the right domain is $\Omega_2 = (0,b)\times(0,2\pi)$ and the interface is $\Omega_I = 0\times(0,2\pi)$. The initial-boundary-value problem (IBVP) we wish to solve is

$$\partial_t u_m = \mathcal{D}_m \Delta u_m + f, \qquad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{54}$$

$$[u_m(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{55}$$

$$[\mathcal{K}_m \partial_x u_m(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{56}$$

$$u_m(\mathbf{x},0) = u^0_m(\mathbf{x}), \qquad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{57}$$

$$u_1(-a,y) = g(-a,y), \quad u_2(b,y) = g(b,y), \quad \text{for } \mathbf{x} \in \Omega_I, \tag{58}$$

where $u_m = u_m(\mathbf{x}, t)$ and we look for solutions that are 2π-periodic in the y-direction. We discretize this problem in time but for clarity keep space continuous. We use the implicit θ-scheme given by

$$\frac{U_m^{n+1}(\mathbf{x}) - U_m^n(\mathbf{x})}{\Delta t} = \theta \mathcal{D}_m \Delta U_m^{n+1}(\mathbf{x}) + (1-\theta)\mathcal{D}_m \Delta U_m^n(\mathbf{x}) + f(\mathbf{x}, t^n), \quad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{59}$$

$$[U_m^{n+1}(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{60}$$

$$[\mathcal{K}_m \partial_x U_m^{n+1}(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{61}$$

$$U_m^0(\mathbf{x}, 0) = u_m^0(\mathbf{x}), \qquad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{62}$$

$$U_1^{n+1}(-a,y) = g(-a,y), \quad U_2^{n+1}(b,y) = g(b,y), \qquad \text{for } \mathbf{x} \in \Omega_I. \tag{63}$$

Here $\Delta t > 0$ is the time step, $U_m^n(\mathbf{x}) \approx u_m(\mathbf{x}, t^n)$, $t^n = n\Delta t$ and $0 < \theta \leqslant 1$. We want to solve the implicit equations (59)–(61), (63) for the solution U_m^{n+1} at the new time level. The interface equations (60) and (61) couple the solutions on the two domains. We can solve these equations by an iteration. We wish to understand the convergence characteristics of this iteration. We first rewrite Eqs. (59)–(61) and (63) as an equation for U_m^{n+1},

$$\Delta U_m^{n+1} - \frac{1}{\theta \mathcal{D}_m \Delta t} U_m^{n+1} = F_m, \qquad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{64}$$

$$[U_m^{n+1}(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{65}$$

$$[\mathcal{K}_m \partial_x U_m^{n+1}(0,y)]_{\mathcal{I}} = 0, \qquad \text{for } \mathbf{x} \in \Omega_I,\ m = 1,2, \tag{66}$$

$$U_1^{n+1}(-a,y) = g(-a,y), \quad U_2^{n+1}(b,y) = g(b,y), \quad \text{for } \mathbf{x} \in \Omega_I. \tag{67}$$

Here $F_m = -[U_m^n/\Delta t + (1-\theta)\mathcal{D}_m \Delta U_m^n(\mathbf{x}) + f(\mathbf{x}, t^n)]/(\theta \mathcal{D}_m)$ is known. Let $U_m^{(j)} \approx U_m^{n+1}$, $j = 0, 1, 2, \ldots$ denote a sequence of iterates and define the following iteration for $j > 0$,

$$\Delta U_m^{(j)} - \frac{1}{\theta \mathcal{D}_m \Delta t} U_m^{(j)} = F_m, \qquad \text{for } \mathbf{x} \in \Omega_m,\ m = 1,2, \tag{68}$$

$$\mathcal{K}_1 \partial_x U_1^{(j)}(0,y) = \mathcal{K}_2 \partial_x U_2^{(j-1)}(0,y), \qquad \text{for } \mathbf{x} \in \Omega_I, \tag{69}$$

$$U_2^{(j)}(0,y) = U_1^{(j)}(0,y), \qquad \text{for } \mathbf{x} \in \Omega_I, \tag{70}$$

$$U_1^{(j)}(-a,y) = g(-a,y), \quad U_2^{(j)}(b,y) = g(b,y), \quad \text{for } \mathbf{x} \in \Omega_I. \tag{71}$$

We first solve for $U_1^{(j)}$ using (68) and the Neumann interface condition (69) and then solve for $U_2^{(j)}$ using (68) and the Dirichlet interface condition (70). To analyze this iteration we proceed in the usual way and first subtract out a particular solution to Eqs. (64) and (63) that satisfies homogeneous Dirichlet conditions at the interface. We also Fourier transform in y (with dual variable k). This results in the following iteration for the functions $W_m^{(j)}(x,k)$, $j = 1, 2, 3, \ldots$,

$$\partial_x^2 W_1^{(j)} = \beta_1^2 W_1^{(j)}, \quad \partial_x^2 W_2^{(j)} = \beta_2^2 W_2^{(j)}, \tag{72}$$

$$\mathcal{K}_1 \partial_x W_1^{(j)}(0,k) = \mathcal{K}_2 \partial_x W_2^{(j-1)}(0,k) + f_I(k), \quad W_2^{(j)}(0,k) = W_1^{(j)}(0,k), \tag{73}$$

$$W_1^{(j)}(-a,k) = 0, \quad W_2^{(j)}(b,k) = 0, \tag{74}$$

where

$$\beta_m^2 = k^2 + \frac{1}{\theta \mathcal{D}_m \Delta t}.$$

The solution to these equations satisfying the boundary conditions (74) is of the form (note that $\beta_m > 0$)

$$W_1^{(j)} = A_j \sinh(\beta_1 (x+a)), \tag{75}$$

$$W_2^{(j)} = B_j \sinh(\beta_2 (x-b)). \tag{76}$$

Substitution of Eqs. (75) and (76) into the interface conditions (73) gives

$$A_j \mathcal{K}_1 \beta_1 \cosh(\beta_1 a) = B_{j-1} \mathcal{K}_2 \beta_2 \cosh(\beta_2 b) + f_I,$$
$$B_j \sinh(-\beta_2 b) = A_j \sinh(\beta_1 a).$$

Whence

$$A_j = -\frac{\mathcal{K}_2}{\mathcal{K}_1} \frac{\beta_2}{\beta_1} \frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)} A_{j-1} + \frac{1}{\beta_1} \mathcal{K}_1 \cosh(\beta_1 a) f_I.$$

This iteration will converge provided the amplification factor, $\mathcal{A}$, defined by

$$\mathcal{A} \equiv -\frac{\mathcal{K}_2}{\mathcal{K}_1}\frac{\beta_2}{\beta_1}\frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)}, \tag{77}$$

$$= -\frac{\mathcal{K}_2}{\mathcal{K}_1}\sqrt{\frac{1/(\theta\mathcal{D}_2\Delta t)+k^2}{1/(\theta\mathcal{D}_1\Delta t)+k^2}}\frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)}, \tag{78}$$

satisfies $|\mathcal{A}| < 1$. For $1/(\theta\mathcal{D}_m\Delta t) \gg k^2$ or $1/(\theta\mathcal{D}_m\Delta t) \ll k^2$ we find that

$$\mathcal{A} \approx \begin{cases} -\frac{\mathcal{K}_2}{\mathcal{K}_1}\sqrt{\frac{\mathcal{D}_1}{\mathcal{D}_2}}\frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)}, & \text{for } 1/(\theta\mathcal{D}_m\Delta t) \gg k^2, \\ -\frac{\mathcal{K}_2}{\mathcal{K}_1}\frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)}, & \text{for } 1/(\theta\mathcal{D}_m\Delta t) \ll k^2. \end{cases} \tag{79}$$

Note that in many cases $\beta_1 a \gg 1$ and $\beta_2 b \gg 1$ in which case $\tanh(\beta_1 a) \approx 1$ and $\tanh(\beta_2 b) \approx 1$ in the above formulae. Then for small wave-numbers (smooth components of the solution), the convergence rate of the interface iteration will be approximated by

$$|\mathcal{A}| \approx \frac{\mathcal{K}_2}{\mathcal{K}_1}\sqrt{\frac{\mathcal{D}_1}{\mathcal{D}_2}}. \tag{80}$$

We see that the convergence rate depends both on the ratio of the thermal conductivities and the square root of the ratio of the thermal diffusivities.

If $|\mathcal{A}| > 1$ then there is no convergence. In this case, however, we can redefine the basic iteration (68)–(71) so that the Neumann interface condition is imposed on $U_2^{(j)}$ and the Dirichlet on $U_1^{(j)}$ and then the new amplification factor is the inverse of $\mathcal{A}$. Thus if $\mathcal{A} \ll 1$ (or $\mathcal{A} \gg 1$) then the basic iteration (or the basic iteration with interface conditions switched) will work well. The difficult case is when $\mathcal{A} \approx 1$ for then the iteration will converge slowly. For this difficult case we can consider acceleration procedures to improve the convergence rate of the iteration. One of the simplest acceleration procedures uses a relaxation parameter. Define a relaxed iteration with relaxation parameter ω by

$$A_j = (1-\omega)A_{j-1} + \omega\left(\mathcal{A}A_{j-1} + \frac{1}{\beta_1\mathcal{K}_1\cosh(\beta_1 a)}f_I\right),$$

in which case

$$A_j = [1-\omega(1+|\mathcal{A}|)]A_{j-1} - \omega\frac{1}{\beta_1\mathcal{K}_1\cosh(\beta_1 a)}f_I.$$

This iteration will converge provided $0 < \omega < 2\omega_{\text{opt}}$ where the optimal value for ω is

$$\omega_{\text{opt}} = \frac{1}{1+|\mathcal{A}|}. \tag{81}$$

Note 1: The under-relaxed iteration can be implemented in practice by adjusting the Neumann interface condition (69), using instead,

$$\mathcal{K}_1\partial_x U_1^{(j)}(0,y) = (1-\omega)\mathcal{K}_1\partial_x U_1^{(j-1)}(0,y) + \omega(\mathcal{K}_2\partial_x U_2^{(j-1)}(0,y)). \tag{82}$$

Note 2: A simple strategy for choosing a value for ω for general interfaces is as follows. Solve the problem with $\omega = 1$ and measure the convergence rate of the residuals in the interface equations. Choose ω from Eq. (81) using this convergence rate in place of $|\mathcal{A}|$.

6.2. The segregated interface (SI) method with mixed interface conditions

We now consider more general segregated interface conditions where instead of applying a Dirichlet interface condition on one domain and a Neumann interface condition on the other, we apply a mixed condition on both sides. Following the development in the previous section, we replace Eqs. (68)–(71) with

$$\Delta U_m^{(j)} - \frac{1}{\theta\mathcal{D}_m\Delta t}U_m^{(j)} = F_m, \quad m = 1,2, \tag{83}$$

$$(a_{1n}\mathcal{K}_1\partial_x + a_{10})U_1^{(j)}(0,y) = (a_{1n}\mathcal{K}_2\partial_x + a_{10})U_2^{(j-1)}(0,y), \tag{84}$$

$$(-a_{2n}\mathcal{K}_2\partial_x + a_{20})U_2^{(j)}(0,y) = (-a_{2n}\mathcal{K}_1\partial_x + a_{20})U_1^{(j)}(0,y), \tag{85}$$

$$U_1^{(j)}(-a,y) = g(-a,y), \quad U_2^{(j)}(b,y) = g(b,y). \tag{86}$$

Note that the Dirichlet–Neumann approach of the previous section corresponds to the choice $a_{1n}=1,\ a_{20}=1$ and $a_{10}=a_{2n}=0$. If the iteration (83)–(86) converges to some values U_m^*, then

$$\begin{bmatrix} a_{1n} & a_{10} \\ -a_{2n} & a_{20} \end{bmatrix} \begin{bmatrix} \mathcal{K}_1\partial_x U_1^* - \mathcal{K}_2\partial_x U_2^* \\ U_1^* - U_2^* \end{bmatrix} = 0. \tag{87}$$

Thus, provided the determinant of the coefficient matrix is non-zero, $a_{1n}a_{20}+a_{10}a_{2n}\neq 0$, it follows that if the iteration converges then both interface conditions will be satisfied. Proceeding as before, we are led to analyze the following iteration

$$\partial_x^2 W_1^{(j)} = \beta_1^2 W_1^{(j)}, \quad \partial_x^2 W_2^{(j)} = \beta_2^2 W_2^{(j)}, \tag{88}$$

$$W_1^{(j)}(-a) = 0, \quad W_2^{(j)} = 0, \tag{89}$$

$$(a_{1n}\mathcal{K}_1\partial_x + a_{10})W_1^{(j)}(0,y) = (a_{1n}\mathcal{K}_2\partial_x + a_{10})W_2^{(j-1)}(0,y) + a_{1n}\hat{f}_I(k), \tag{90}$$

$$(-a_{2n}\mathcal{K}_2\partial_x + a_{20})W_2^{(j)}(0,y) = (-a_{2n}\mathcal{K}_1\partial_x + a_{20})W_1^{(j)}(0,y) - a_{2n}\hat{f}_I(k). \tag{91}$$

The solution to these equations is also of the form (75) and (76) and the amplification factor for the mixed interface conditions is

$$\mathcal{A} = \frac{a_{1n}\mathcal{K}_2\beta_2 - a_{10}\tanh(\beta_2 b)}{a_{1n}\mathcal{K}_1\beta_1 + a_{10}\tanh(\beta_1 a)} \frac{a_{2n}\mathcal{K}_1\beta_1 - a_{20}\tanh(\beta_1 a)}{a_{2n}\mathcal{K}_2\beta_2 + a_{20}\tanh(\beta_2 b)}. \tag{92}$$

We note then that if $a_{1n}=a_{2n}=a_{10}=a_{2n}=\frac{1}{2}$ then

$$\mathcal{A} = \frac{\mathcal{K}_1\beta_1 - \tanh(\beta_1 a)}{\mathcal{K}_1\beta_1 + \tanh(\beta_1 a)} \frac{\mathcal{K}_2\beta_2 - \tanh(\beta_2 b)}{\mathcal{K}_2\beta_2 + \tanh(\beta_2 b)}, \tag{93}$$

so that $|\mathcal{A}|<1$ and the iteration will always converge (although the convergence rate may not be very good). Although the choice $a_{10}=a_{1n}\mathcal{K}_2\beta_2/\tanh(\beta_2 b)$ and $a_{20}=a_{2n}\mathcal{K}_1\beta_1/\tanh(\beta_1 a)$ would make $\mathcal{A}=0$, this is not practical since $\beta_m=\beta_m(k)$ depends on wave-number k. After some numerical experimentation we have arrived at the following choice for the coefficients of the mixed conditions that seems to give reasonable results,

$$a_{1n}=1, \quad a_{2n}=1, \quad \begin{cases} a_{10}=\gamma_2^2/\gamma_1, \quad a_{20}=\gamma_1^2, & \text{if } \gamma_1 \geqslant \gamma_2, \\ a_{10}=\gamma_2^2, \quad a_{20}=\gamma_1^2/\gamma_2, & \text{if } \gamma_1 < \gamma_2, \end{cases} \tag{94}$$

where

$$\gamma_m = \mathcal{K}_m\beta_m(k_m^*) = \mathcal{K}_m\sqrt{\frac{1}{\theta\mathcal{D}_m\Delta t} + (k_m^*)^2}, \tag{95}$$

and $(k_m^*)^2 = 0.1/(\mathcal{D}_m\Delta t),\ m=1,2$, is a rough guess of a wave-number where the errors are significant. This form is motivated by the fact that when $\gamma_1 \gg \gamma_2$ or $\gamma_2 \gg \gamma_1$ the mixed condition will be close to the Dirichlet–Neumann condition and should converge rapidly. When $\gamma_1 \approx \gamma_2$ and $\gamma_1 \geqslant \gamma_2$ the iteration should also converge rapidly since

$$\mathcal{A} \approx -\frac{\mathcal{K}_2\beta_2(k) - \mathcal{K}_2\beta_2(k_2^*)\tanh(\beta_2 b)}{\mathcal{K}_1\beta_1(k) + \mathcal{K}_2\beta_2(k_2^*)\tanh(\beta_1 a)} \frac{\tanh(\beta_1 a)}{\tanh(\beta_2 b)}. \tag{96}$$

A similar result holds when $\gamma_1 \approx \gamma_2$ and $\gamma_1 < \gamma_2$. In Section 6.3, we present some numerical results using the mixed interface condition with parameters chosen following (94).

6.3. Results for implicit time-stepping and interfaces

In this section we present computational results that provide confirmation of the analyzes of the previous sections. We solve some conjugate heat transfer problems using the segregated interface method and measure the convergence rates of the iterations. These convergence rates are compared to the theoretical values. We consider using the Dirichlet–Neumann interface condition (DN) discussed in Section 6.1, as well as using a mixed interface condition (M) described in Section 6.2.

From Eq. (80) we define an estimated convergence rate of the DN approach,

$$\sigma_{\text{est}} = \frac{\mathcal{K}_2}{\mathcal{K}_1}\sqrt{\frac{\mathcal{D}_1}{\mathcal{D}_2}}.$$

This is the convergence rate that might be expected when a Neumann condition is applied at the interface to sub-domain Ω_1, with parameters $\mathcal{K}_1,\ \mathcal{D}_1$ and a Dirichlet condition is applied at the interface to sub-domain Ω_2, with parameters $\mathcal{K}_2,\ \mathcal{D}_2$.

We begin by considering a domain consisting of two adjacent squares, $\Omega_s=\Omega_1\cup\Omega_2$ where $\Omega_1=(-1,0)\times(0,1)$ and $\Omega_2=(0,1)\times(0,1)$. We define a grid $\mathcal{G}_s$ for Ω_s using Cartesian grids on each domain with grid spacing equal to $\Delta x=1/320$. We solve the heat equation in each sub-domain with a second-order accurate implicit predictor–corrector method where the corrector step is given by the implicit θ-scheme (59) with $\theta=\frac{1}{2}$ and $\Delta t=.01$. The boundary conditions

case	grid	IC	$(\mathcal{D}_1,\mathcal{K}_1)$	$(\mathcal{D}_2,\mathcal{K}_2)$	σ_{est}	σ_{comp}	ω	$\sigma_{\text{comp}}(\omega)$
0	two-squares	DN	(1.0, 1.0)	(0.1, 0.1)	.32	.31	.76	.08
1		DN	(0.5, 1.0)	(0.1, 0.2)	.45	.44	.69	.12
2		DN	(1.0, 1.0)	(1.0, 1.0)	1.0	1.0	.5	.19
3		M	(1.0, 1.0)	(1.0, 1.0)	1.0	.45	.69	.65
4		M	(1.0, 1.0)	(0.1, 0.1)	.32	.17	.85	.13
5	disk-in-a	DN	(1.0, 1.0)	(1.0, 1.0)	1.0	1.2	.5	.11
6	square	DN	(0.5, 1.0)	(0.1, 0.2)	.45	.48	.69	.04
7		DN	(1.0, 1.0)	(1.0, .02)	.02	.024	.98	.0013
8		M	(1.0, 1.0)	(1.0, 1.0)	1.0	.04	.96	.015
9		M	(0.5, 1.0)	(0.1, 0.2)	.45	.21	.83	.0013
10		M	(1.0, 1.0)	(1.0, .02)	.02	.022	.98	.00056

Fig. 7. Convergence rates for solving the interface equations by iteration with the segregated interface approach and weakly coupled implicit time-stepping. The interface condition (IC) is either Dirichlet–Neumann (DN) or mixed (M). σ_{est} is the estimated convergence rate for the DN condition. σ_{comp} is the computed convergence rate. ω is the relaxation parameter computed from σ_{comp} using Eq. (81). $\sigma_{\text{comp}}(\omega)$ is the convergence rate of the relaxed iteration using this value of ω.

on non-interface boundaries are taken as Dirichlet (similar results are obtained when Neumann boundary conditions are used). The exact solution, determined with the method of analytic solutions (Section 7.1) is a fourth-degree polynomial. At each time-step we solve the interface equations by the iteration (68)–(71). An initial guess for the right-hand-side to the interface equations is determined by extrapolation in time from previous values. We estimate the average convergence rate, σ_{comp}, from the reduction in the residual in the interface equations over a total of N iterations,

$$\sigma_{\text{comp}} = \left(\frac{R^{(N)}}{R^{(1)}}\right)^{1/N},$$
$$R^{(j)} \equiv \max_{\mathbf{i},\mathbf{j}} \left\{ |U^{(j)}_{1,\mathbf{i}} - U^{(j)}_{2,\mathbf{j}}|, |\mathcal{K}_1 D_n U^{(j)}_{1,\mathbf{i}} - \mathcal{K}_2 D_n U^{(j)}_{2,\mathbf{j}}| \right\},$$

where the maximum is taken over all grid values on the interface and D_n is the discrete approximation to the normal derivative. We solve the interface equations until $R^{(N)} < \epsilon$ where $\epsilon = 10^{-10}$ for the computations given in this section. The values of σ_{comp} are then averaged over a few time steps to give the tabulated result. Fig. 7 shows some results for different values of $(\mathcal{K}_m, \mathcal{D}_m)$. The estimated convergence rate σ_{est} is generally a good approximation to the computed value σ_{comp} for cases that use the DN conditions. For example, in case "1" the estimated converge rate is $\sigma_{\text{est}} = 0.45$ and the computed is $\sigma_{\text{comp}} = 0.44$. We also show the convergence rate when a relaxation parameter, $\omega \approx 1/(1+\sigma_{\text{comp}})$, is used (82). In some of these cases, the convergence rate at each iteration, $R^{(j+1)}/R^{(j)}$, can be quite variable with the initial iterations tending to converge much faster than later iterations. This can be attributed to the smooth components of the error converging rapidly on the first few iterations, while high-frequency modes of the error become more important for later iterations. Fig. 7 also shows results for the disk-in-a-square grid, $\mathcal{G}^{(8)}$, defined in Section 7.2. The estimated and computed convergence rates for the DN interface condition agree well in this case as well. The mixed interface condition converges very rapidly for case "8", $\mathcal{D}_1 = 1,\ \mathcal{K}_1 = 1,\ \mathcal{D}_2 = 1$ and $\mathcal{K}_2 = 1$ which is the difficult case for the DN approach. The mixed condition generally gives good results. Case "3" for the mixed condition on the two-squares problem does not converge as well as case "8" for the disk-in-a-square problem. The convergence rates for case "3" start out very small but then slow down due to boundary effects where the interface meets the adjacent top and bottom boundaries. In case "8" the interface is a periodic circle and thus this effect does not appear. It may be possible to remedy this problem with an improved approximation for points near where the interface meets another boundary.

7. Numerical results

In this section we present some computational results. Although we are primarily concerned with demonstrating the accuracy of the multi-domain algorithms, we will also show results of applying the approach to a few interesting conjugate heat transfer problems. In all cases we solve the incompressible Navier–Stokes equations (2) in the fluid domains and the heat equation (4) in the solid domains. These equation are solved together with appropriate initial conditions, boundary conditions as defined in Section 3.1, and interface conditions (5) and (6). We will present results using explicit time-stepping and the coupled interface (CI) method as well as results for implicit time-stepping and the segregated interface (SI) method.

For explicit time-stepping we use a predictor–corrector method that consists of a second-order Adams–Bashforth predictor step followed by a second-order Adams–Moulton corrector step (trapezoidal rule). For implicit time-stepping we use a semi-implicit version of the explicit predictor–corrector method. The corrector-step takes the form of the θ-scheme (59). In the semi-implicit approach we can optionally treat different terms in the equations in a implicit manner. For the INS equations, for example, we may only treat the viscous terms implicitly or alternatively treat both viscous and advection terms in an implicit manner.

We now list the different test cases and describe their purpose:

(i) The *solid disk in a fluid* example of Section 7.2 is a relatively simple two-dimensional fluid–solid problem that is verified with the method of analytic solutions as defined in Section 7.1. This geometry is also used in Section 6.3 for verifying convergence rates of the (SI) method.
(ii) The *flat-plate heat exchanger with buoyancy* test case of Section 7.3 defines an exact solution in a simple geometry and shows that our approximations give the exact answer for solutions that are cubic polynomials in space, when solved on Cartesian grids.
(iii) The *flat-plate heat exchanger with the method of analytic solutions* example of Section 7.4 is used to verify that the numerical solution is exact, up to round-off errors, when the true solution is a second-degree polynomial in space and time. This case shows that the explicit (CI) and implicit (SI) time-stepping algorithms are second-order accurate in time.
(iv) The *conjugate heat transfer in cocentric cylinders* problem of Section 7.5 defines a three-dimensional exact solution in a curvilinear geometry and thus tests the implementation for curvilinear three-dimensional grids.
(v) The *fluid in a curved pipe* test case of Section 7.6 provides a more general three-dimensional curvilinear grid example that is verified with the method of analytic solutions. This geometry is also used to perform the simulations that provide the parallel scaling data of Section 7.7.
(vi) The *conjugate heat transfer in an hohlraum* problem of Section 7.8 provides a relatively complex and realistic example involving multiple fluid domains and multiple solid domains. This case demonstrates that we can solve problems in complex geometries and that we can handle many domains, including multiple fluid domains.
(vii) The *conjugate heat transfer in an hexagonal fuel-assembly* example of Section 7.9 shows that we can solve problems in complex three-dimensional domains and provides further parallel scaling results.

7.1. The method of analytic solutions

The *method of analytic solutions* is an extremely useful technique for constructing exact solutions to check the accuracy of a numerical implementation. This method, also sometimes known as the *method of manufactured solutions* [51], or *twilight-zone forcing* [2] adds forcing functions to the governing equations and boundary conditions. These forcing functions are determined so that some given functions, $\bar{u}(\mathbf{x},t)$, will be the exact solution to the forced equations. With this approach, the error in the discrete solution can be easily determined. As an example of the technique, consider solving the IBVP for the advection–diffusion equation,

$$\begin{aligned} &u_t + \mathbf{a}\cdot\nabla u - \nu\Delta u = f, \quad \text{for } \mathbf{x}\in\Omega,\\ &u(x,0) = u_0(\mathbf{x}), \quad \text{for } \mathbf{x}\in\Omega, \text{ at } t=0,\\ &u(\mathbf{x},t) = g(\mathbf{x},t), \quad \text{for } \mathbf{x}\in\partial\Omega. \end{aligned}$$

Any given smooth function, $\bar{u}(\mathbf{x},t)$, will be an exact solution of the IBVP if we set the forcing function, initial conditions and boundary conditions as

$$f(\mathbf{x},t) = \bar{u}_t + \mathbf{a}\cdot\nabla\bar{u} - \nu\Delta\bar{u}, \quad u_0(\mathbf{x}) = \bar{u}(\mathbf{x},0) \quad \text{and} \quad g(\mathbf{x},t) = \bar{u}(\mathbf{x},t).$$

In our numerical implementation, we have a number of choices available for $\bar{u}$, including polynomials, trigonometric functions and exponential functions, among others. The exact form of the analytic solution we use in each case will be given in subsequent sections.

7.2. Solid disk in a fluid

We consider a conjugate heat transfer problem for a heated solid disk in a fluid. We solve the problem in a two-dimensional domain $\Omega = \Omega_S \cup \Omega_F$, where the solid domain Ω_S consists of a circular disk of radius $R = 0.4$ and the fluid domain Ω_F is a square that surrounds the fluid, $\Omega_F = [-1,1]^2 - \Omega_S$. The grid for an annular region is defined by

$$\mathcal{A}([r_a,r_b],N_1,N_2) = \{(r_{i_2}\cos(\theta_{i_1}), r_{i_2}\sin(\theta_{i_1}))\,|\,\theta_{i_1} = 2\pi i_1/N_1,\ r_{i_2} = r_a + (r_b - r_a)i_2/N_2,\ i_k = 0,1,\ldots,N_k,\ k=1,2\}.$$

The grid for a rectangle is

$$\mathcal{R}([x_a,x_b]\times[y_a,y_b],N_1,N_2) = \{(x_a + (x_b - x_a)i_1/N_1, y_a + (y_b - y_a)i_2/N_2)\,|\,i_k = 0,1,\ldots,N_k,\ k=1,2\}.$$

The number of grid points in each coordinate direction for a grid with resolution factor j is chosen so that the grid spacing is approximately

$$\Delta s^{(j)} = \frac{1}{20j}.$$

The composite grid for the solid domain consists of an inner square grid and an outer annulus,

$$\mathcal{G}_S^{(j)} = \mathcal{R}([-R_i - \Delta s^{(j)}, R_i + \Delta s^{(j)}]^2, N_{x,S}, N_{x,S}) \cup \mathcal{A}([R_i, R], N_\theta, N_r),$$

where $R = 0.4$, $R_i = R - 0.175$, $N_{x,S} = \lfloor 2(R_i + \Delta s^{(j)})/\Delta s^{(j)} + 1.5 \rfloor$, $N_\theta = \lfloor 2\pi R/\Delta s^{(j)} + 1.5 \rfloor$ and $N_r = \lfloor (R - R_i)/\Delta s^{(j)} + 2.5 \rfloor$. Here $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. The composite grid for the fluid domain is composed from a background square and an annular grid,

$$\mathcal{G}_F^{(j)} = \mathcal{R}([-1, 1]^2, N_{x,F}, N_{x,F}) \cup \mathcal{A}([R, R + 0.175], N_\theta, N_r),$$

where $N_{x,F} = \lfloor 2/\Delta s^{(j)} + 1.5 \rfloor$. The composite grid for the entire multi-domain problem is the union of the fluid and solid grids,

$$\mathcal{G}^{(j)} = \mathcal{G}_F^{(j)} \cup \mathcal{G}_S^{(j)}.$$

Fig. 8 shows the grid $\mathcal{G}^{(1)}$ for this two-domain problem.

We solve the incompressible Navier–Stokes (INS) equations in the fluid domain and the heat equation in the solid domain. We use the method of analytic solutions as described in Section 7.1 with a trigonometric exact solution. The exact solution for the fluid is

$$\bar{u} = \frac{1}{2}\cos(f_x \pi x)\cos(f_y \pi y)\cos(f_t \pi t), \tag{97}$$

$$\bar{v} = \frac{1}{2}\sin(f_x \pi x)\sin(f_y \pi y)\cos(f_t \pi t), \tag{98}$$

$$\bar{p} = \cos(f_x \pi x)\sin(f_y \pi y)\cos(f_t \pi t), \tag{99}$$

$$\overline{T} = \cos(f_x \pi x)\cos(f_y \pi y)\cos(f_t \pi t), \tag{100}$$

which satisfies $\nabla \cdot (\bar{u}, \bar{v}) = 0$ if $f_x = f_y$. For the fluid we choose $f_x = f_y = f_t = 1$. For the solid the exact solution for T is also given by (100) and we choose $f_x = f_y = f_t = 2$. Since the exact solutions for T in the fluid and solid do not match at the interface, the interface jump conditions (5) and (6) are replaced by $[T]_{\mathcal{I}} = [\overline{T}]_{\mathcal{I}}$ and $[\mathcal{K}\partial_n T]_{\mathcal{I}} = [\mathcal{K}\partial_n \overline{T}]_{\mathcal{I}}$. We use parameters values $\nu = 0.025$, $\mathcal{D}_f = 0.03$ and $\mathcal{K}_f = 0.04$ for the fluid, and $\mathcal{D}_s = 0.04$, $\mathcal{K}_s = 0.9$ for the solid, with the coefficient of thermal expansivity $\alpha = 0.1$ and the gravity vector equal to $\mathbf{g} = (0, -10)$. Fig. 8 shows the computed solution for the temperature at time $t = 1$. We define the maximum error in a computed solution $U_{\mathbf{i}}^n$ on grid $\mathcal{G}^{(j)}$ as

$$e_u^{(j)} = \max_{\mathbf{i} \in \mathcal{G}^{(j)}} |U_{\mathbf{i}}^n - \bar{u}(\mathbf{x}_{\mathbf{i}}, t^n)|, \tag{101}$$

where the maximum is taken over all valid points on the composite grid $\mathcal{G}^{(j)}$. The convergence rate σ for a component u of the solution is estimated by assuming that the maximum norm of the error has the form $e_u^{(j)} = C(\Delta s^{(j)})^\sigma$ and then making a least-squares fit for σ to the equation $\log(e_u^{(j)}) = \sigma \log(\Delta s^{(j)}) + \log(C)$ for different values of $\Delta s^{(j)}$.

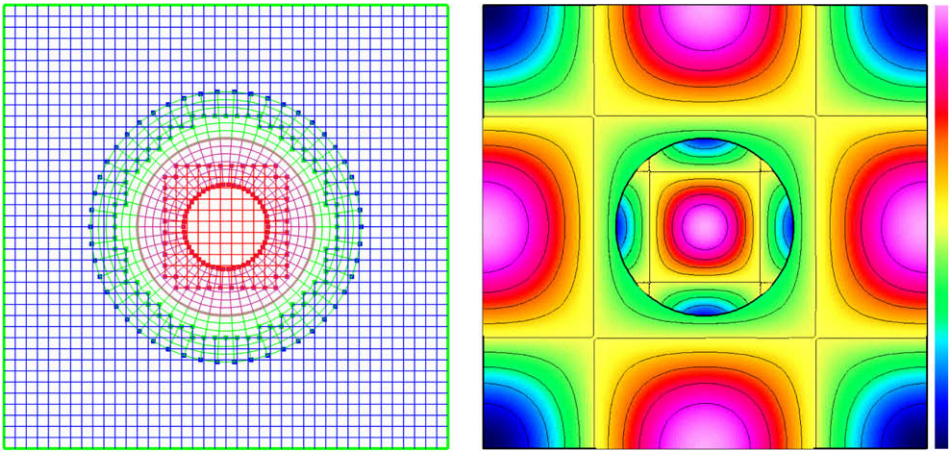

Fig. 8. Left : composite grid $\mathcal{G}^{(1)}$ for the two-domain conjugate heat transfer problem. The heat equation is solved in the inner disk and the incompressible Navier–Stokes equations are solved in the outer region. Right : the computed temperature for a problem where the exact solution was constructed using the method of analytic solutions.

We solve the problem with explicit time-stepping and the coupled interface approach. We also solve with implicit time-stepping and the segregated interface approach. Figs. 9 and 10 present the maximum errors at time $t = 1$ and the estimated convergence rates when using explicit and implicit time-stepping, respectively. For the implicit time-stepping case we note that the time-step Δt was explicitly reduced by a factor of two as the grids were refined by the same factor so that the time-stepping errors would be reduced appropriately. For the SI approach, the results when using the Dirichlet–Neumann interface conditions or the mixed interface conditions are nearly the same. The convergence results provide strong evidence that the overall scheme is second-order accurate.

7.3. A flat-plate heat exchanger with buoyancy

In this example we consider the buoyancy driven flow past a flat-plate for which an exact steady state solution is available. Fig. 11 shows the two-dimensional (2D) and three-dimensional (3D) computational domains. For the two-dimensional case the fluid occupies the upper rectangle $\Omega^F = [x_a, x_b][y_i, y_b]$ and a solid occupies the lower rectangle $\Omega^S = [x_a, x_b][y_a, y_i]$. The boundary conditions and initial conditions are

$$
\begin{aligned}
&T(x, y_a, t) = T_a, \quad T(x, y_b, t) = T_b, \quad \text{for } x_a \leqslant x \leqslant x_b,\\
&\mathbf{u}(x, y_i, t) = 0, \quad \mathbf{u}(x, y_b, t) = 0, \quad \text{for } x_a \leqslant x \leqslant x_b,\\
&\mathbf{u}(x, y, 0) = 0, \quad \text{for } \mathbf{x} \in \Omega^F, \quad T(x, y, 0) = \frac{1}{2}(T_a + T_b), \quad \text{for } \mathbf{x} \in \Omega^F \cup \Omega^S.
\end{aligned}
$$

Gravity is in the x-direction, $\mathbf{g} = (g_x, 0, 0)$ and the solution is assumed to be periodic in x. The steady state solution is

$$
\begin{aligned}
T_\infty(y) &= \begin{cases} T_a + \dfrac{y - y_a}{y_i - y_a}(T_i - T_a), & \text{for } y_a \leqslant y \leqslant y_i,\\ T_i + \dfrac{y - y_i}{y_b - y_i}(T_b - T_i), & \text{for } y_i \leqslant y \leqslant y_b, \end{cases}\\
u_\infty(y) &= \frac{\alpha g_x}{\nu}\left(\frac{T_a + T_b}{4}(y - y_i)(y - y_b) + \frac{T_b - T_a}{6(y_b - y_i)}(y - y_i)(y - y_b)(y - y_m)\right), \quad \text{for } y_i \leqslant y \leqslant y_b,
\end{aligned}
$$

where $T_i = \dfrac{\mathcal{K}_f T_b + \mathcal{K}_s T_a}{\mathcal{K}_f + \mathcal{K}_s}$, and $y_m \equiv \dfrac{y_i + y_b}{2}$.

Note that the velocity profile is perturbed from standard plane Poiseuille flow with the maximum in the velocity shifted towards the hotter wall. We solve the problem with parameters

		Fluid					Solid
Grid $\mathcal{G}^{(j)}$	$\Delta s^{(j)}$	$e_p^{(j)}$	$e_u^{(j)}$	$e_v^{(j)}$	$e_T^{(j)}$	$\nabla \cdot \mathbf{u}$	$e_T^{(j)}$
$\mathcal{G}^{(1)}$	1/20	8.0×10^{-3}	4.7×10^{-3}	8.1×10^{-3}	4.6×10^{-3}	2.6×10^{-2}	4.6×10^{-3}
$\mathcal{G}^{(2)}$	1/40	2.2×10^{-3}	1.1×10^{-3}	1.8×10^{-3}	1.4×10^{-3}	7.5×10^{-3}	1.4×10^{-3}
$\mathcal{G}^{(4)}$	1/80	6.0×10^{-4}	2.6×10^{-4}	3.9×10^{-4}	3.8×10^{-4}	9.0×10^{-4}	3.8×10^{-4}
rate, σ		1.87	2.09	2.20	1.80	2.43	1.80

Fig. 9. Maximum errors at $t = 1$ and estimated convergence rate, σ, when solving the *solid disk in a fluid* problem using explicit time-stepping and the coupled interface (CI) approach.

		Fluid					Solid
Grid $\mathcal{G}^{(j)}$	$\Delta s^{(j)}$	$e_p^{(j)}$	$e_u^{(j)}$	$e_v^{(j)}$	$e_T^{(j)}$	$\nabla \cdot \mathbf{u}$	$e_T^{(j)}$
$\mathcal{G}^{(1)}$	1/20	1.4×10^{-2}	9.3×10^{-3}	1.2×10^{-2}	4.7×10^{-3}	1.2×10^{-2}	4.7×10^{-3}
$\mathcal{G}^{(2)}$	1/40	2.5×10^{-3}	$1.$				

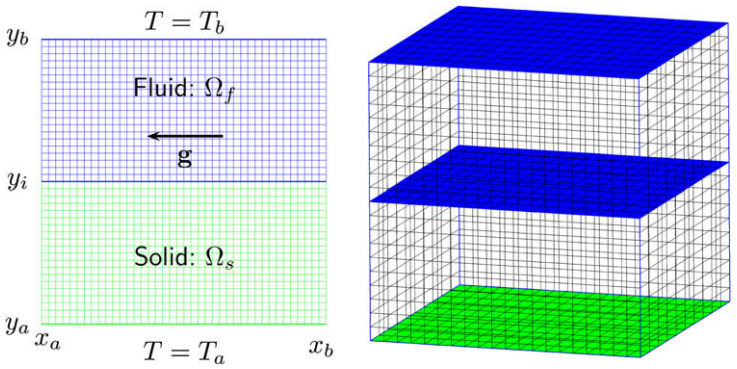

Fig. 11. Flat-plate heat exchanger geometry and grids for the two-dimensional and three-dimensional computations. A rectangular fluid domain sits above a rectangular solid domain.

$$x_a = 0, \quad x_b = 1, \quad y_a = 0, \quad y_i = \frac{1}{2}, \quad y_b = 1, \quad \nu = 0.05, \quad \alpha = 1, \quad g_x = -1,$$
$$\mathcal{K}_s = 1, \quad \mathcal{D}_s = 1, \quad \mathcal{K}_f = 0.2, \quad \mathcal{D}_f = 0.2. \tag{102}$$

The solution is integrated with implicit time-stepping with the segregated interface approach until $t = 10$ when the steady state has been approximately reached. The numerical solution remains independent of x to round-off error. The computed solution is compared to the exact solution in Fig. 12 along the line $x = \frac{1}{2}$ for a grid with grid spacing $\Delta x = \Delta y = 1/40$. Since the exact solution of the temperature is a linear profile, and the exact solution to the velocity is a cubic, the computed solution on a Cartesian grid is equal to the exact solution to within round-off error or the degree to which the solution has converged to a steady state. Note that the reason the cubic profile can be computed exactly is due to the fact that the leading-order truncation-error term in approximating u_{xx} with a second-order accurate finite difference approximation on a Cartesian grid is $(\Delta x^2/12)\partial_x^4 u$.

Fig. 13 shows the corresponding results for the three-dimensional problem for a grid with grid spacing $\Delta x = \Delta y = \Delta z = 1/20$. The numerical solutions again match the exact solution to round-off errors as in the two-dimensional case.

7.4. A flat-plate heat exchanger with the method of analytic solutions

In this section we consider the same flat-plate geometry as in the previous section. We solve the conjugate heat transfer problem on this domain using the method of analytic solutions. We choose the exact solution to be a degree two polynomial in space times a degree two polynomial in time. We show that we can compute the exact solution to this problem using explicit time-stepping and the CI method or using implicit time-stepping and the SI method. For implicit time-stepping we use the second-order accurate Crank–Nicolson (trapezoidal) rule in each domain. This problem is an excellent test of the implementation since it can identify subtle mistakes that would otherwise be difficult to find. This test also verifies that

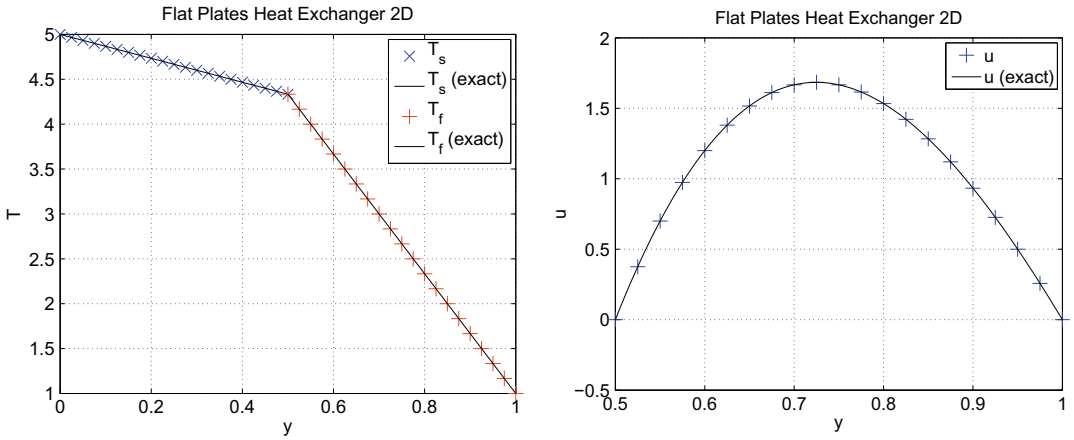

Fig. 12. Results for the 2D flat-plate heat exchanger. The temperature is a piecewise linear profile and the fluid velocity is a cubic polynomial. The numerical solution computed on a Cartesian grid gives the *exact* solution up to round-off.

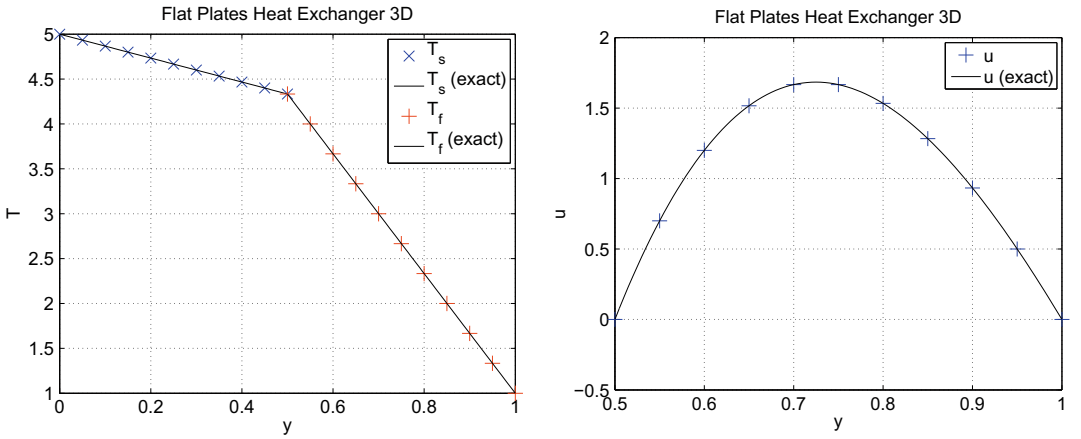

Fig. 13. Results for the 3D flat-plate heat exchanger. The temperature is a piecewise linear profile and the fluid velocity is a cubic polynomial. The numerical solution computed on a Cartesian grid gives the *exact* solution up to round-off.

the time-stepping algorithms are second-order accurate in time. We remark that the spatial errors in the scheme usually dominate the overall error. By choosing a problem where the spatial errors are negligible we can carefully test the time accuracy.

The exact solution for the fluid is given by

$$\bar{u} = (x^2 + 2xy + y^2)(1 + t/2 + t/3), \tag{103}$$

$$\bar{v} = (x^2 - 2xy - y^2)(1 + t/2 + t/3), \tag{104}$$

$$\bar{p} = (x^2 + y^2 - 1 + xy/2)(1 + t/2 + t/3), \tag{105}$$

$$\overline{T} = \left(\frac{1}{2} + x/2 + y/4 + x^2/3 + y^2/6\right)(1 + t/2 + t/3), \tag{106}$$

where $\nabla \cdot (\bar{u}, \bar{v}) = 0$. The exact solution for the solid is

$$\overline{T} = (2 + x + y/2 + x^2/2 + y^2/4)(1 + t/2 + t/3).$$

The fluid and solid parameters were chosen as in Eq. (102). The boundary conditions for the fluid were chosen as no-slip walls on all boundaries. The boundary conditions for the solid where chosen as Dirichlet. For the implicit time-stepping case using the CI method the final errors do depend on the tolerance that we set for solving the interface equations. We used a tolerance of 10^{-13}. In Fig. 14 we indicate the maximum errors in the solution at time $t = 1.0$ for explicit and implicit time-stepping. The results show that the solution is exact up to round-off errors. Although not shown, similar results are obtained in three-dimensions with the solution being exact up to round-off errors.

7.5. Conjugate heat transfer in cocentric cylinders

In this example we consider the buoyancy driven conjugate heat transfer between two cocentric cylinders as shown in Fig. 15. A solid occupies the inner hollow cylindrical domain $\Omega^S = \{\mathbf{x} | r \in [r_a, r_i],\ y \in [y_a, y_b]\}$ where $r = \sqrt{x^2 + z^2}$ and the axial direction is parallel to the y-axis. A fluid occupies the outer hollow cylindrical domain $\Omega^F = \{\mathbf{x} | r \in [r_i, r_b],\ y \in [y_a, y_b]\}$. The boundary conditions and initial conditions for the problem under consideration are

	Fluid					Solid
Time Stepping	e_p	e_u	e_v	e_T	$\nabla \cdot \mathbf{u}$	e_T
explicit (CI)	8.4×10^{-15}	1.8×10^{-15}	1.8×10^{-15}	1.0×10^{-14}	1.8×10^{-14}	9.8×10^{-15}
implicit (SI)	1.1×10^{-13}	2.2×10^{-14}	1.1×10^{-14}	7.5×10^{-14}	2.3×10^{-13}	7.6×10^{-14}

Fig. 14. Maximum errors at $t = 1$ for the flat-plate heat exchanger solved using the method of analytic solutions with explicit and implicit time-stepping. The exact solution is a polynomial of degree 2 in space times a polynomial of degree 2 in time. The numerical solution is exact up to round-off errors.

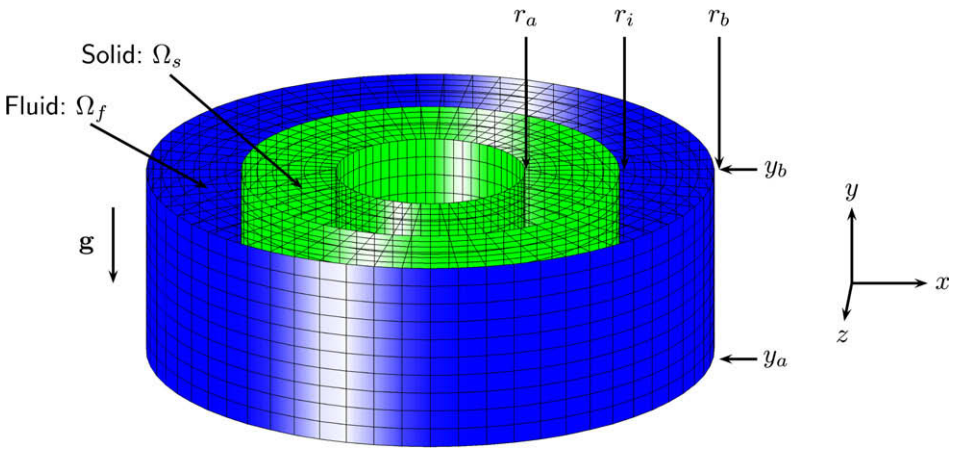

Fig. 15. Cocentric cylinders heat exchanger geometry and grids. The coarse grid $\mathcal{G}^{(1)}$ is shown.

$$
\begin{aligned}
&T(\mathbf{x},t) = T_a, && \text{for } r = r_a,\ y_a \leqslant y \leqslant y_b,\\
&\mathbf{u}(\mathbf{x},t) = 0, && \text{for } r = r_i,\ y_a \leqslant y \leqslant y_b,\\
&T(\mathbf{x},t) = T_b, \quad \mathbf{u}(\mathbf{x},t) = 0, && \text{for } r = r_b,\ y_a \leqslant y \leqslant y_b,\\
&\mathbf{u}(\mathbf{x},0) = 0, \quad T(\mathbf{x},0) = \frac{1}{2}(T_a + T_b), && \text{for } \mathbf{x} \in \Omega^F,\\
&T(\mathbf{x},0) = \frac{1}{2}(T_a + T_b), && \text{for } \mathbf{x} \in \Omega^S.
\end{aligned}
$$

Gravity is in the y-direction, $\mathbf{g} = (0, g_y, 0)$ and the solution is assumed to be periodic in the axial direction y. The steady state solution is a function of r only and is given by

$$
T_\infty(r) = \begin{cases} T_a + \dfrac{T_i - T_a}{\ln(r_i/r_a)} \ln(r/r_a), & \text{for } r_a \leqslant r \leqslant r_i,\\ T_i + \dfrac{T_b - T_i}{\ln(r_b/r_i)} \ln(r/r_i), & \text{for } r_i \leqslant r \leqslant r_b, \end{cases}
$$

$$
v_\infty(r) = \frac{\alpha g_y}{4\nu}((c_1 - c_2)r^2 + (c_2 r^2 + c_3)\ln(r) + c_4), \quad \text{for } r_i \leqslant r \leqslant r_b,
$$

where

$$
\begin{aligned}
T_i &= \frac{\mathcal{K}_f \ln(r_i/r_a) T_b + \mathcal{K}_s \ln(r_b/r_i) T_a}{\mathcal{K}_f \ln(r_i/r_a) + \mathcal{K}_s \ln(r_b/r_i)},\\
c_2 &= \frac{T_b - T_i}{\ln(r_b/r_i)}, \quad c_1 = T_i - c_2 \ln(r_i),\\
c_3 &= \frac{(c_1 - c_2)(r_i^2 - r_b^2) + c_2(r_i^2 \ln(r_i) - r_b^2 \ln(r_b))}{\ln(r_b/r_i)},\\
c_4 &= \frac{(c_1 - c_2)(r_b^2 \ln(r_i) - r_i^2 \ln(r_b)) + c_2 \ln(r_i)\ln(r_b)(r_b^2 - r_i^2)}{\ln(r_b/r_i)}.
\end{aligned}
$$

We solve the problem with parameters

$$
\begin{aligned}
&r_a = \frac{1}{2}, \quad r_i = 1, \quad r_b = \frac{3}{2}, \quad y_a = 0, \quad y_b = 1, \quad T_a = 5, \quad T_b = 1,\\
&\nu = 0.05, \quad \alpha = 1, \quad g_y = -1, \quad \mathcal{K}_s = 1, \quad \mathcal{D}_s = 1, \quad \mathcal{K}_f = 0.2, \quad \mathcal{D}_f = 0.2.
\end{aligned}
$$

A grid for a cylindrical region with axial direction parallel to the y-axis is defined by

$$
\begin{aligned}
\mathcal{C}([r_a, r_b], [y_a, y_b], N_1, N_2, N_3) = \{&(r_{i_2}\cos(\theta_{i_1}), y_a + (y_b - y_a) i_3/N_3, r_{i_2}\sin(\theta_{i_1})) \,|\, \theta_{i_1} = 2\pi i_1/N_1,\ r_{i_2} = r_a + (r_b - r_a) i_2/N_2,\\
&i_k = 0, 1, \ldots, N_k,\ k = 1, 3\}.
\end{aligned}
$$

The number of grid points is chosen so that the grid spacing is approximately

$$
\Delta s^{(j)} = \frac{1}{10j}.
$$

The composite grid for two-domain problem consists of two cylindrical grids,

$$\mathcal{G}^{(j)} = \mathcal{C}([r_a, r_i], [y_a, y_b], N_\theta, N_r, N_y) \cup \mathcal{C}([r_i, r_b], [y_a, y_b], N_\theta, N_r, N_y),$$

where $N_\theta = \lfloor 2\pi r_i/\Delta s^{(j)} + 1.5 \rfloor$, $N_r = \lfloor (r_i - r_a)/\Delta s^{(j)} + 2.5 \rfloor$ and $N_y = \lfloor (y_b - y_a)/\Delta s^{(j)} + 1.5 \rfloor$. We solve the problem on a sequence of grids $\mathcal{G}^{(j)}$, $j = 1, 2, 4$. The solution is integrated with implicit time-stepping with the segregated interface approach until $t = 10$ when the steady state has been approximately reached.

The computed solution is compared to the exact solution in Fig. 16 along the radial line $y = \frac{1}{2}$, $z = 0$, for the grid $\mathcal{G}^{(2)}$. The numerical solution shows excellent agreement with the exact solution. Fig. 17 shows the maximum errors in T and v and the estimated convergence rate. The results demonstrate that the method is second-order accurate.

7.6. Fluid in a curved pipe

We consider the solution to a conjugate heat transfer problem in a curved pipe consisting of an inner fluid region surrounded by a solid pipe as shown in Fig. 18. We solve this problem on a sequence of grids with the method of analytic solutions and determine the errors and convergence rates when using explicit and implicit time-stepping. These results show the accuracy of the three-dimensional capabilities of our approach. We also solve a more realistic conjugate heat transfer problem.

For completeness, we first define the problem domain and the grids. The fluid domain for the curved pipe is a 90° sector of a toroid. The solid domain is a toroidal shell that forms the solid pipe. The grid for a sector of a toroidal shell is defined as a body of revolution of an annulus cross-section and is given by

$$\mathcal{T}(\mathbf{x}_c, [r_a, r_b], [\phi_a, \phi_b], R, N_1, N_2, N_3) = \{\mathbf{x}_c + ((R + r_{i_2}\cos(\theta_{i_1}))\cos(\phi_{i_3}), r_{i_2}\sin(\theta_{i_1}), (R + r_{i_2}\cos(\theta_{i_1}))\sin(\phi_{i_3})) \\ | \theta_{i_1} = 2\pi i_1/N_1,\ r_{i_2} = r_a + (r_b - r_a)i_2/N_2,\ \phi_{i_3} = \phi_a + (\phi_b - \phi_a)i_3/N_3,\ i_k = 0, 1, \ldots, N_k,\ k = 1, 3\},$$

where $\mathbf{x}_c$ is the center of the toroid, r_a and r_b are the inner and outer values for the minor radius, R is the major radius and ϕ_a and ϕ_b are the bounds on the toroidal angle ϕ. The grid for the core of the fluid domain is a body of revolution of a square cross-section and given by

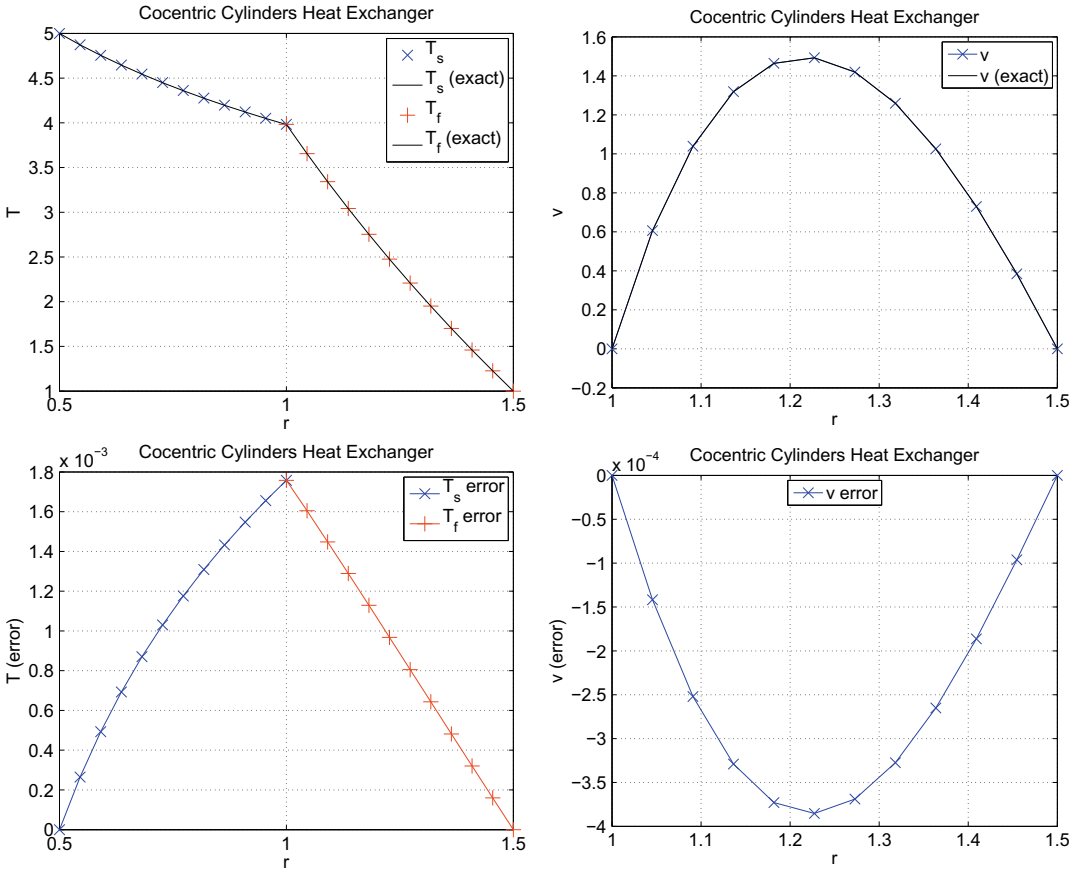

Fig. 16. Results for the cocentric cylinder heat exchanger for grid $\mathcal{G}^{(2)}$. Left: the computed solution, exact solution and errors for the temperature in the solid and fluid domains along a radial line. Right: the computed solution, exact solution and errors for the velocity component v along a radial line.

Grid $\mathcal{G}^{(j)}$	$\Delta s^{(j)}$	$e_v^{(j)}$	$e_T^{(j)}$
$\mathcal{G}^{(1)}$	1/10	1.3×10^{-3}	6.0×10^{-3}
$\mathcal{G}^{(2)}$	1/20	3.9×10^{-4}	1.8×10^{-3}
$\mathcal{G}^{(4)}$	1/40	9.4×10^{-5}	4.5×10^{-4}
rate, σ		1.9	1.9

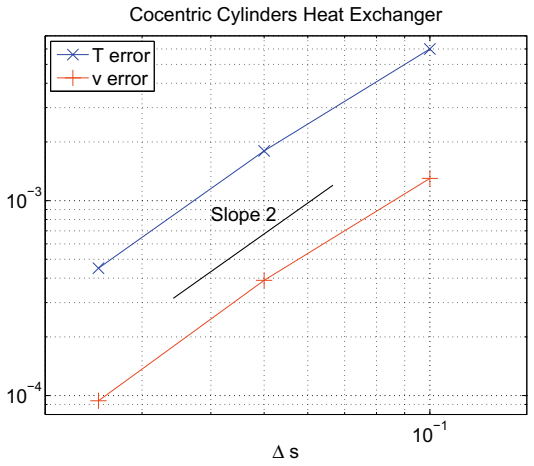

Fig. 17. Maximum errors and estimated convergence rate for the numerical solution of the steady state cocentric cylinders heat exchanger.

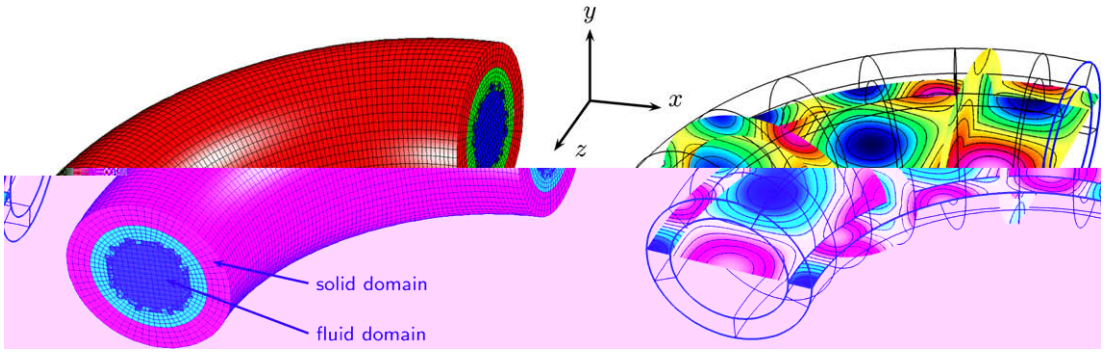

Fig. 18. Conjugate heat transfer of a fluid in a curved solid pipe with an exact solution constructed with the method of analytic solutions. Left : the composite grid $\mathcal{G}^{(1)}$ for the two-domain problem. The inner fluid region is discretized with the green and blue grids; the outer solid region with the red grid. Right: the computed solution for the temperature in the fluid and solid domains. Contours of the solution are plotted on planes that cut through the domain. The fluid and solid domains use different trigonometric functions as exact solutions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\mathcal{S}(\mathbf{x}_c,[x_a,x_b]\times[y_a,y_b],[\phi_a,\phi_b],R,N_1,N_2,N_3)=\{\mathbf{x}_c+((R+x_{i_1})\cos(\phi_{i_3}),y_a+(y_b-y_a)i_2/N_2,(R+x_{i_1})\sin(\phi_{i_3}))\,|\,$$
$$x_{i_1}=x_a+(x_b-x_a)i_1/N_1,\ \phi_{i_3}=\phi_a+(\phi_b-\phi_a)i_3/N_3,\ i_k=0,1,\ldots,N_k,\ k=1,2\}.$$

The composite grid for the fluid domain consists of a toroidal shell and inner core,

$$\mathcal{G}_F^{(j)}=\mathcal{T}\Big((R,0,0),[0.35,0.5],[\pi,\pi/2],2.5,N_\theta^{(j)},N_{fr}^{(j)},N_\phi^{(j)}\Big)$$
$$\cup\,\mathcal{S}\Big((R,0,0),[-0.35,0.35]\times[-0.35,0.35],[\pi,\pi/2],2.5,N_x^{(j)},N_y^{(j)},N_\phi^{(j)}\Big),$$

while that for the solid domain consists of the outer toroidal shell,

$$\mathcal{G}_S^{(j)}=\mathcal{T}\Big((R,0,0),[0.5,0.7],[\pi,\pi/2],2.5,N_\theta^{(j)},N_{sr}^{(j)},N_\phi^{(j)}\Big).$$

Here j denotes the resolution of the grids. The number of grid points, $N_\theta^{(j)}$, $N_{fr}^{(j)}$, $N_\phi^{(j)}$, etc. are chosen so that the grid spacing is approximately equal to

$$\Delta s^{(j)}=\frac{1}{20j}.$$

The composite grid for the entire multi-domain problem is the union of the fluid and solid grids,

$$\mathcal{G}^{(j)}=\mathcal{G}_F^{(j)}\cup\mathcal{G}_S^{(j)}.$$

We solve the incompressible Navier–Stokes equations with Boussinesq approximation in the fluid domain and the heat equation in the solid domain. We use the method of analytic solutions with a trigonometric exact solution. The exact solution for the fluid is

$$\bar{u} = \cos(f\pi x)\cos(f\pi y)\cos(f\pi z)\cos(f_t\pi t), \tag{107}$$

$$\bar{v} = \frac{1}{2}\sin(f\pi x)\sin(f\pi y)\cos(f\pi z)\cos(f_t\pi t), \tag{108}$$

$$\bar{w} = \frac{1}{2}\sin(f\pi x)\cos(f\pi y)\sin(f\pi z)\cos(f_t\pi t), \tag{109}$$

$$\bar{p} = \cos(f\pi x)\cos(f\pi y)\cos(f\pi z)\cos(f_t\pi t), \tag{110}$$

$$\overline{T} = \cos(f\pi x)\cos(f\pi y)\cos(f\pi z)\cos(f_t\pi t), \tag{111}$$

which satisfies $\nabla\cdot(\bar{u},\bar{v},\bar{w}) = 0$. For the fluid we choose $f = f_t = 1.5$. For the solid the exact solution for T is also given by (111) and we choose $f = f_t = 1$. Since the exact solutions for T in the fluid and solid do not match at the interface, the interface jump conditions become $[T]_{\mathcal{I}} = [\overline{T}]_{\mathcal{I}}$ and $[\mathcal{K}\partial_n T]_{\mathcal{I}} = [\mathcal{K}\partial_n \overline{T}]_{\mathcal{I}}$. We use parameters values $\nu = 0.0125$, $\mathcal{D}_f = \nu/0.72$ and $\mathcal{K}_f = 0.1$ for the fluid and $\mathcal{D}_s = 0.025$, $\mathcal{K}_s = 0.05$ for the solid.

Fig. 18 shows the computed solution for the temperature at time $t = 1$. Note that the trigonometric functions were chosen so that the solution for T in the solid is different from T in the fluid. Figs. 19 and 20 present the maximum errors at time $t = 1$ and the estimated convergence rates when using explicit and implicit time-stepping. The results indicate that the solution is converging at rates close to second-order accuracy. The actual errors between the explicit and implicit time-stepping results are similar.

For the implicit time-stepping case, the interface equations were solved to a tolerance of 10^{-3} in the maximum residuals of the interface jump conditions. This tolerance was always achieved within two iterations. The interface equations will always be solved for at least two iterations with the implicit predictor–corrector time-stepping method since they are solved after the predictor step and after the first corrector-step. Thus, in this example no additional corrector steps were needed to satisfy the interface equations.

To give an idea of what the solution looks like for a more realistic situation we solve a conjugate heat transfer problem through the curved pipe including the effects of buoyancy. We compare the two cases when gravity points in the $+z$ and $-z$ directions. For the fluid, an inflow condition of $(u,v,w) = (0,0,-1)\mathcal{P}(\mathbf{x})$ and $T = 0$ is specified at the boundary face at $z = 0$. Here the *parabolic inflow* function $\mathcal{P}(\mathbf{x})$ is given by

$$\mathcal{P}(\mathbf{x}) = \begin{cases} 1 & \text{for dist}(\mathbf{x}) > \delta, \\ 1 - (\text{dist}(\mathbf{x})/\delta)^2 & \text{for dist}(\mathbf{x}) \leqslant \delta, \end{cases}$$

where dist($\mathbf{x}$) is the distance of a point $\mathbf{x}$ on the inflow face to the adjacent solid wall, and δ is chosen to be 0.2. An outflow condition is specified at the face at $x = 2.5$ using $p_n + p = 0$, while no-slip walls are used on the curved boundaries of the pipes. (Section 3.1 provides more information on the equations used for the different boundary conditions.) The solid pipe

		Fluid						Solid
Grid $\mathcal{G}^{(j)}$	$\Delta s^{(j)}$	$e_p^{(j)}$	$e_u^{(j)}$	$e_v^{(j)}$	$e_w^{(j)}$	$e_T^{(j)}$	$\nabla\cdot\mathbf{u}$	$e_T^{(j)}$
$\mathcal{G}^{(1)}$	1/20	9.1×10^{-3}	3.7×10^{-3}	2.4×10^{-3}	3.7×10^{-3}	4.8×10^{-3}	1.1×10^{-1}	3.9×10^{-3}
$\mathcal{G}^{(2)}$	1/40	2.2×10^{-3}	6.5×10^{-4}	5.8×10^{-4}	7.5×10^{-4}	1.1×10^{-3}	3.5×10^{-2}	9.5×10^{-4}
rate, σ		2.06	2.50	2.03	2.30	2.14	1.73	2.05

Fig. 19. Maximum errors at $t = 1$ and estimated convergence rate, σ, when solving the *fluid in a curved pipe* problem using explicit time-stepping and the coupled interface (CI) approach. See also Fig. 21.

		Fluid						Solid
Grid $\mathcal{G}^{(j)}$	$\Delta s^{(j)}$	$e_p^{(j)}$	$e_u^{(j)}$	$e_v^{(j)}$	$e_w^{(j)}$	$e_T^{(j)}$	$\nabla\cdot\mathbf{u}$	$e_T^{(j)}$
$\mathcal{G}^{(1)}$	1/20	1.2×10^{-2}	5.6×10^{-3}	2.6×10^{-3}	4.7×10^{-3}	5.9×10^{-3}	1.2×10^{-1}	4.5×10^{-3}
$\mathcal{G}^{(2)}$	1/40	2.9×10^{-3}	8.9×10^{-4}	6.0×10^{-4}	9.2×10^{-4}	1.4×10^{-3}	3.4×10^{-2}	1.0×10^{-3}
rate, σ		2.02	2.66	2.08	2.35	2.10	1.81	2.09

Fig. 20. Maximum errors at $t = 1$ and estimated convergence rate, σ, when solving the *fluid in a curved pipe* problem using implicit time-stepping and the segregated interface (SI) approach. See also Fig. 21.

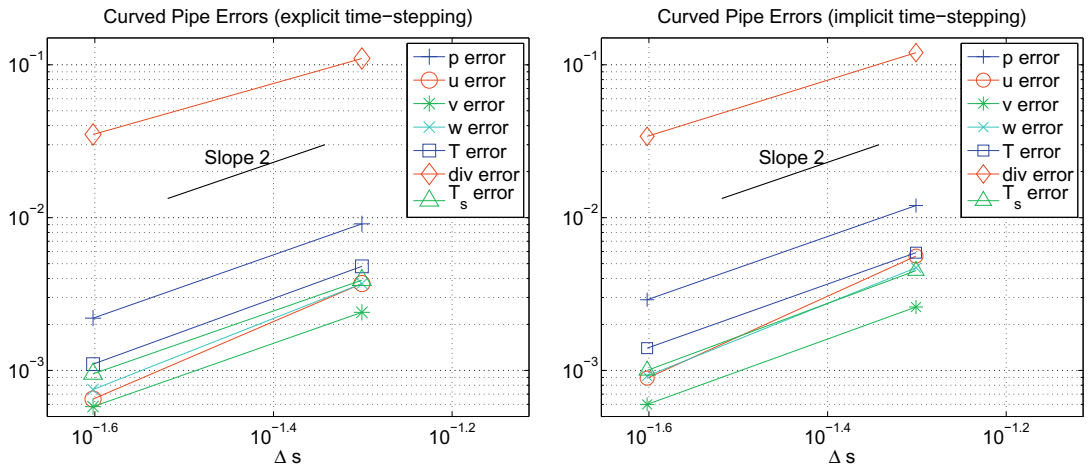

Fig. 21. Maximum errors at $t = 1$ when solving the *fluid in a curved pipe* problem using explicit time-stepping and the coupled interface (CI) approach (left) and using implicit time-stepping and the segregated interface (SI) approach (right). See also Figs. 19 and 20.

is given a fixed temperature $T = 10$ on the outer boundary of the pipe and Neumann conditions on the inflow and outflow faces. Other parameters are taken as $\nu = 0.025$, $\mathcal{D}_f = 0.025/0.72$, $\mathcal{K}_f = 0.1$, $\mathcal{D}_s = 0.05$, $\mathcal{K}_s = 1$. Gravity is chosen as $\mathbf{g} = (0, 0, \pm 1)$ for two different cases. The initial conditions for the fluid were based on the state $(u, v, w) = (0, 0, -1)$ and $T = 0$. These values for the velocity were projected and smoothed to provide smooth and approximately divergence free initial conditions. The initial condition for the solid was taken as $T = 10$.

Fig. 22 shows the solution at time $t = 1$ for the two cases of gravity pointing in the $+z$ and $-z$ directions. These solutions were computed on the grid $\mathcal{G}^{(2)}$. The temperature in the fluid and solid are shown along with the speed of the flow, $|\mathbf{u}| = \sqrt{u^2 + v^2 + w^2}$. The fluid is heated as it moves through the pipe. There are significant differences in the flow between the two cases. In the case when gravity points in the $-z$ direction, the buoyancy force will be in the $+z$ direction for hot fluid. Near the inflow region this will slow down the fluid near the walls. When gravity is in the $+z$ direction, the buoyancy force will act in the opposite direction on hot fluid. In this case the hot fluid near the wall in the inflow section is accelerated.

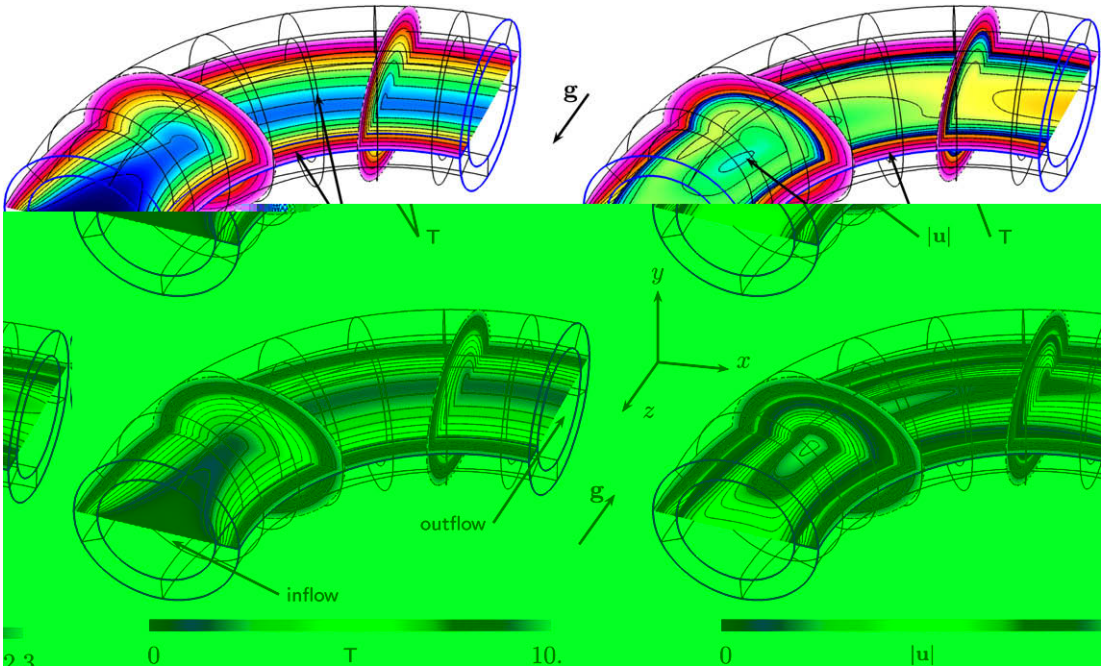

Fig. 22. Flow of a heated incompressible fluid through a curved solid pipe. Left : the temperature in the fluid and solid. Right: the flow speed (magnitude of the velocity) and the temperature in the solid. Top: gravity is $\mathbf{g} = (0, 0, +1)$. Bottom: gravity is $\mathbf{g} = (0, 0, -1)$.

7.7. Parallel performance

The multi-domain solver runs in parallel on distributed memory computers. In this section we provide some parallel performance numbers when solving the flow in a curved pipe example given in the later part of Section 7.6. Our approach to distributed memory parallelism and load balancing is described in [14]. In brief, each grid can be independently partitioned across the processors, while a modified bin-packing algorithm is used for load balancing. We solve the flow in a curved pipe problem on the composite grid $\mathcal{G}^{(2)}$ for which the total number of grid points was approximately 4.6×10^5. We solve to time $t = 0.4$ (513 time steps) with explicit time-stepping. The pressure equation is solved with a stabilized bi-conjugate gradient algorithm from PETSc [52] using an ILU (incomplete LU) preconditioner with fill level 1 and a relative tolerance of 10^{-5}. On average, approximately nine iterations were required per pressure solve. The solution to the pressure equation was taking over 50% of the total time.

For each run, we integrate the equations from $t = 0$ to 0.4 and record the number of time steps taken and the total CPU time used (wall clock time). From this information we compute $\mathcal{T}_k$, the average CPU time per step for run k. Let $N_{\text{proc}}^{(k)}$ denote the number of processors used in run k. To measure the parallel scaling behaviour, we define a *parallel scaling factor*

$$\mathcal{S}_k = \frac{\mathcal{T}_0}{\mathcal{T}_k} \frac{N_{\text{proc}}^{(0)}}{N_{\text{proc}}^{(k)}},$$

which compares the CPU times per step between runs 0 and k. The run for $k = 0$ is taken to be a reference computation with one processor. Ideally, $\mathcal{S}_k$ should equal 1 for perfect scaling. All calculations in this section were performed on a AMD Opteron Linux cluster with eight 2.4 GHz processors per node and 16 gigabytes of memory per node. Fig. 23 presents the strong parallel scaling results. Shown are the parallel scaling factor and the time-per-step in CPU seconds for the total computation, the initialization phase of the computation and the advance stage of the computation. The initialization phase includes the time to setup the pressure equation and project the initial conditions to be approximately divergence free. The time to solve

k	Nodes	processors	total (s/step)	init (s/step)	advance (s/step)	S_k	S_k/S_{k-1}
0	1	1	7.2	.50	6.7	1.0	–
1	2	2	4.5	.31	4.2	.80	.80
2	2	4	2.4	.26	2.1	.75	.94
3	2	8	1.3	.12	1.2	.70	.93
4	4	8	1.3	.11	1.2	.70	1.0
5	2	16	.71	.065	.64	.63	.90
6	4	32	.39	.036	.36	.58	.92
7	8	64	.20	.026	.18	.56	.97

Fig. 23. Strong parallel scaling for flow of a fluid in a curved pipe. The solution was solved on grid $\mathcal{G}^{(2)}$ with approximately 4.6×10^5 grid points for 513 steps. S_k is the parallel scaling factor. See also Fig. 24.

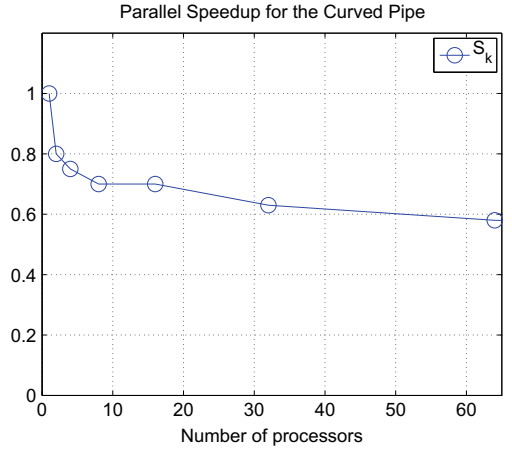

Fig. 24. Parallel speedup, S_k, for the curved pipe. See also Fig. 23.

the coupled interface equations was small, typically 3–5% of the total time. There is a decrease in the parallel scaling factor of $S_0 = 1.0$ to $S_1 = 0.8$ in going from 1 to 2 processors but after that the code scales fairly well out to 64 processors. These results are quite good considering the relatively few number of grids points for this problem. Fig. 24 provides a graph of S_k versus the number of processors.

7.8. Conjugate heat transfer in an hohlraum

In this section we illustrate the use of our multi-domain simulation capability in solving an interesting conjugate heat transfer problem. Inertial confinement fusion (ICF) targets used by the National Ignition Facility (NIF) contain a spherical shell of deuterium–tritium (DT) ice. This shell is enclosed by a capsule, typically made of a polymer or metal, and suspended within a thin walled gold cylinder known as a hohlraum. Fig. 25(a) depicts a typical target and hohlraum configuration. Since spherical symmetry of the ice layer directly affects the performance of the target, the temperature on the ice shell must be kept as uniform as possible. Decay heat released from the DT vapor and ice leads to gravity driven thermal convection both in the DT vapor and the transfer gas between the hohlraum and capsule. This convection leads to asymmetries in the temperature of the ice layer and ultimately to asymmetric sublimation of the ice. One method for actively controlling the temperature in the target includes adding heating and cooling elements at the periphery of the hohlraum. This system can be simulated both with and without the thermal control mechanisms to examine the effectiveness of such approaches.

The five domains used for this problem include three solid domains for the DT ice, capsule and the hohlraum and two fluid domains for the DT vapor and transfer gas. Following Sanchez and Geidt, the capsule is assumed to be a polyimide shell with an outer radius of 1 mm and a thickness of 0.16 mm; the ice shell is 0.08 mm thick. The overall length and radius of the hohlraum are 9.5 mm and 2.75 mm, respectively, with a 0.1 mm thick wall [53]. Material properties for each domain are given in Fig. 26 [53–55]. The transfer gas properties are based on a 50–50 mixture of H_2 and He at 19.5 K. The acceleration due to gravity is a constant 9.8×10^3 mm/s^2. Cryogenic cooling regions at either end of the hohlraum are held at a constant temperature of 19.5 K while the ring heaters provide a heat flux of 1.3 mW/mm^2. Fig. 25(b) illustrates the grids used to mesh this domain; for the purposes of the figure, a coarsened version of the grid is shown.

The results from a sample axisymmetric computation are given in Fig. 27 which shows the temperature in the different domains as well as the streamlines of the flow in the two gas domains. The large buoyancy driven recirculation flow in the

Fig. 25. A typical NIF hohlraum target consists of DT vapor and ice; a polyimide capsule; the heat transfer gas; and the gold hohlraum. The composite grid consists of two fluid and three solid domains using a total of 20 grids.

	ρ	C	$\mathcal{K}$	q	μ
units	$\frac{mg}{mm^3}$	$\frac{mJ}{mgK}$	$\frac{mJ}{s\ mm\ K}$	$\frac{mJ}{(s\ mm^3)}$	$\frac{mg}{mm\ s}$
DT vapor	3.0×10^{-4}	2.49	8.0×10^{-2}	5.0×10^{-5}	1.25×10^{-3}
DT ice	1×10^{-2}	1.0	0.294	4.9×10^{-2}	(solid)
Capsule	1.43	1.15	0.15	0.0	(solid)
Transfer gas	1.25×10^{-3}	10.5	2.2×10^{-2}	0.0	2.7×10^{-3}
Hohlraum	19.3	0.13	1×10^{3}	0.0	(solid)

Fig. 26. Material properties used for the hohlraum convection problem: ρ is the density, C the specific heat, $\mathcal{K}$ the thermal conductivity, q the decay heat source, and μ the dynamic viscosity.

transfer gas region is clearly evident as well as the asymmetric temperature distribution. There is also a recirculation flow in the inner DT vapor region although the magnitude of the velocity is much smaller in this region. These results are in qualitative agreement with those previously reported [53]. We leave to future work a more detailed analysis of this problem and comparison to experiments.

7.9. Conjugate heat transfer in an hexagonal fuel-assembly

As a final example we consider the flow of a fluid coolant past a collection of hot fuel pins arranged in an hexagonal assembly. Such a geometry is similar to those found in some sodium-cooled fast nuclear reactors [56]. The computational domains and grids are shown in Fig. 28. There are three different domains. The seven cylindrical fuel pins define one solid domain. The hexagonal solid duct defines a second solid domain. The fluid channel occupies the domain between the fuel pins and the outer duct.

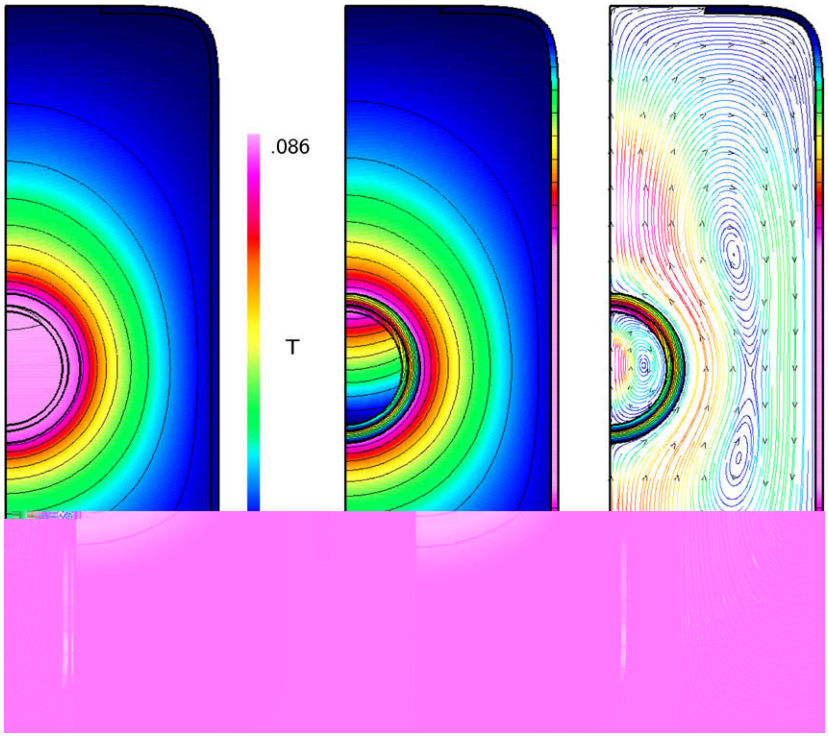

Fig. 27. Hohlraum results. Left: the temperature in each of the domains shown on the same scale. Middle: the temperature is shown with different scales in each domain to better show the variation. For the middle plot the DT vapor has bounds on T of [0.085, 0.086], the DT ice [0.084, 0.086], the capsule [0.081, 0.086], the transfer gas [0, 0.083] and hohlraum [0, 0.005]. Right: the stream lines in the central DT gas region and transfer gas region. The flow speed in DT vapor has bounds of $[0, 1.7 \times 10^{-4}]$ while the bounds in the transfer gas are [0, 0.81].

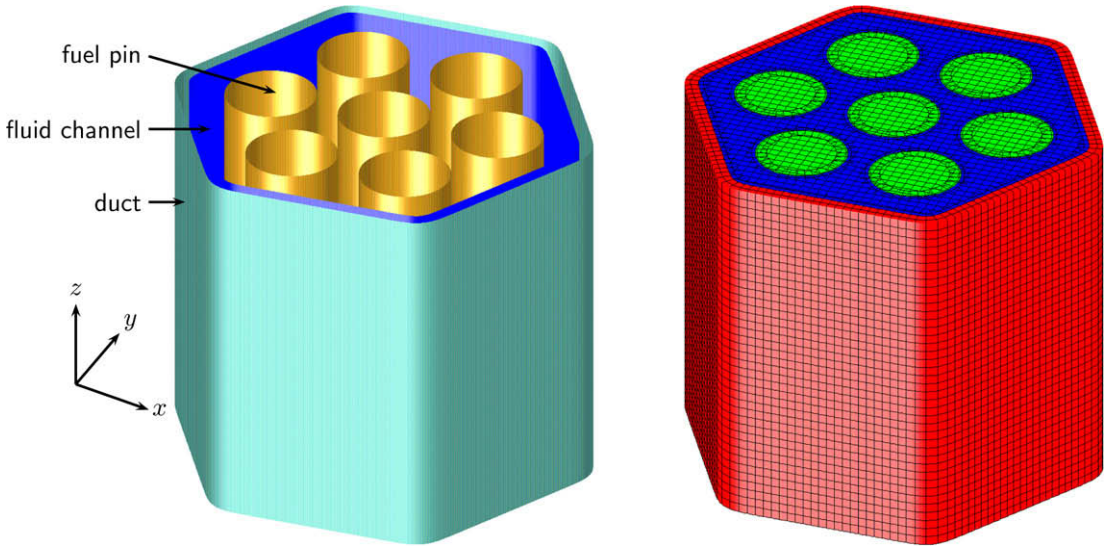

Fig. 28. The fuel-assembly geometry and composite grid. A coarsened version of the grid is shown. Grids for the fluid domain are shown in blue, those for the fuel pins in green, while the grid for the duct is shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The composite grid for this domain was defined in terms of cylindrical grids, Cartesian box grids and an extruded *smooth-polygon* grid for the duct. The *smooth-polygon* is defined in terms of logarithms of hyperbolic cosine functions [57]. The fuel pins of radius 0.5 and length 4.0 were separated by a minimum distance of 0.3 between each other and the duct. The composite grid for this domain with resolution factor j, denoted by $\mathcal{G}^{(j)}$, was chosen to have a background grid spacing of $\Delta s^{(j)} = 1/(20j)$. The cylindrical grids in the fluid domain were clustered near the walls to better resolve the fluid boundary layer.

A cold fluid of temperature $T = 0$ enters the domain on the lower inflow boundary at $z = 0$ and travels upward. An outflow condition is imposed at the top boundary at $z = 4$. The boundary condition at the fluid–solid interface is a no-slip wall. The boundaries of the pins and duct that are not on the interface with the fluid channel are taken as adiabatic walls (Neumann boundary conditions). The fuel pins are at an initial temperature of $T = 0$ and given a constant volume heat source $f = 1$ in Eq. (4). The duct is at an initial temperature of $T = 0$. The fluid parameters are taken as $\nu = 0.025$, $\mathcal{D} = 0.0694$, $\mathcal{K} = 0.05$ and $\alpha = 0.1$ The solid fuel pin parameters are $\mathcal{D} = 0.1$, and $\mathcal{K} = 0.5$. The solid duct parameters are $\mathcal{D} = 0.1$, and $\mathcal{K} = 1.0$. The gravity vector is $\mathbf{g} = (0, 0, -1)$.

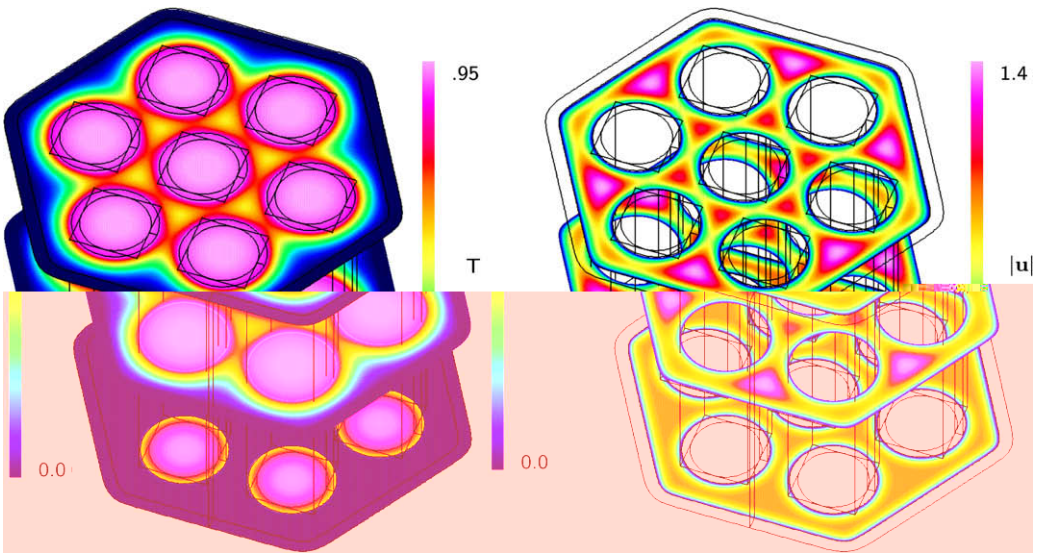

Fig. 29. Conjugate heat transfer in the fuel-assembly. Left: the temperature in the fluid and solid domains plotted on three planes passing through the domain. Right: the flow speed in the fluid domain.

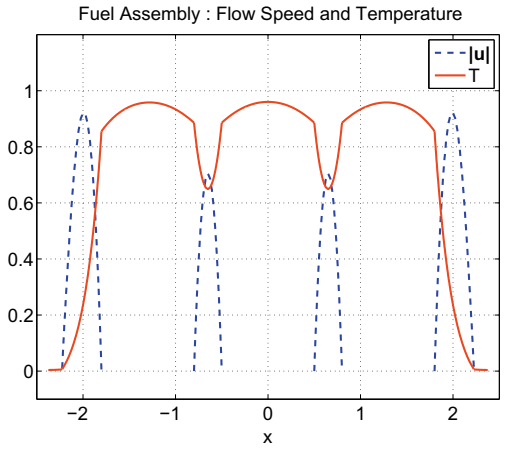

Fig. 30. Conjugate heat transfer in the fuel-assembly. The flow speed and temperature are plotted along a line near the outflow that passes through 3 pins.

The solution was advanced in time with implicit time-stepping and the (SI) interface method. A semi-implicit treatment of the INS equations was used where only the viscous terms were treated implicitly; this required one scalar implicit system that was used for all components of the velocity and a second scalar implicit system for the temperature. The implicit systems and the pressure equation were solved with PETSc [52] using a stabilized bi-conjugate gradient algorithm with an ILU(1) preconditioner [52]. The solution was computed on grid $\mathcal{G}^{(2)}$ which had a total of about 13 million grid points. The solution was computed in parallel using 8 nodes and 64 processors. The CPU time was about 26 s/step for 68 steps requiring a total time of about 30 min. The maximum memory required per processor was about 720 Mb. Running the same problem on 16 nodes and 128 processors took about 18 min for a speedup of about 1.7 going from 64 to 128 processors. Fig. 29 shows the computed solution at $t = 1.0$. Both the temperature and the flow speed, $|\mathbf{u}|$, are shown. From the figure it can be seen that the fuel pins heat the fluid as it flows upward past the pins. The fluid moves faster through the larger sub-channels near the duct. Fig. 30 shows a line plot of the flow speed, $|\mathbf{u}|$, and temperature along the line segment $-2.5 \leqslant x \leqslant 0.25,\ y = 0,\ z = 4$, near the outlet. The temperature is seen to be continuous at the interfaces but as expected the normal derivative in the temperature jumps at the interfaces.

8. Conclusions

We have described an approach for solving transient and steady-state conjugate heat transfer problems in complex geometries using composite overlapping grids. Separate physics solvers for fluid flow and heat conduction are used in different domains. The solutions are coupled at interfaces by conditions that impose the continuity of temperature and heat flux. A multi-domain solver coordinates the overall time-stepping method and the treatment of the interfaces. An analysis of a *centered* approximation to the interface jump condition was given and the approximation was shown to be second-order accurate and stable, independent of the relative sizes of the thermal conductivities and diffusivities. When used with explicit time-stepping, this results in an efficient, strongly coupled algorithm (i.e. no iterations between sub-domains are required to satisfy the interface conditions). We also analyzed iteration strategies for solving the interface conditions when the sub-domain solutions were advanced in a partitioned fashion with implicit time-stepping. Conditions for the convergence when using the Dirichlet–Neumann interface approach were given. The use of a mixed (Robin) interface condition was shown to have attractive convergence properties especially for the case when adjacent materials have similar properties.

The accuracy of the new multi-domain approach was verified on a number of test cases. The method was shown to give excellent results for a flat-plate heat exchanger example and a cylindrical heat exchanger test case for which the exact solutions could be determined. The method was shown to be second-order accurate in space and time using the method of analytic solutions when applied to a two-dimensional heated disk in a square, the flat-plate heat exchanger and the three-dimensional flow in a solid curved pipe. The multi-domain approach was also applied to two interesting applications. The gas flow and heat conduction in an axisymmetric model of a hohlraum were computed. This problem involved two distinct fluid domains coupled to three solid domains. Finally the flow of a coolant fluid in a model of a nuclear reactor fuel-assembly was computed.

There are a variety of future directions for this work. The interface treatment can be extended to higher-order accuracy and to the case when the grids on either side of the interface do not match. A straight-forward extension will enable the solution to conjugate heat transfer problems involving compressible fluid flows using, for example, our *cgcns* solver. The approach can also be extended to treat moving rigid bodies following the technique described in [5]. Another future direction will be to couple fluid flow and deforming solids.

Acknowledgments

This work was performed under the auspices of the US Department of Energy (DOE) by Lawrence Livermore National Laboratory in part under Contract W-7405-Eng-48 and in part under Contract DE-AC52-07NA27344 and by DOE contracts from the ASCR Applied Math Program and the ITAPS SciDAC Center.

Appendix A. Proof of theorems

Here we provide proofs to Theorems 1 and 2.

Theorem 1. *The solution to the one-dimensional interface problem (35)–(41) is stable and second-order accurate.*

Proof. Laplace transforming Eqs. (35)–(41) and replacing (38) with the equivalent condition $U_{1,0}(t) = U_{2,0}(t)$ gives

$$s\widehat{U}_{1,j} - u_1^0(x_{1,j}) = \mathcal{D}_1 D_+ D_- \widehat{U}_{1,j}, \quad \text{for } j = 0,-1,-2,\ldots, \tag{A.1}$$

$$s\widehat{U}_{2,j} - u_2^0(x_{2,j}) = \mathcal{D}_2 D_+ D_- \widehat{U}_{2,j}, \quad \text{for } j = 0,1,2,\ldots, \tag{A.2}$$

$$\mathcal{K}_1 D_0 \widehat{U}_{1,0} = \mathcal{K}_2 D_0 \widehat{U}_{2,0}, \tag{A.3}$$

$$\widehat{U}_{1,0} = \widehat{U}_{2,0}. \tag{A.4}$$

$$\|\widehat{U}_1\|_h < \infty, \quad \|\widehat{U}_2\|_h < \infty, \tag{A.5}$$

where $\widehat{U}_{m,j}(s)$ is the Laplace transform of $U_{m,j}(t)$ with dual variable s. The Laplace transform of the error in the approximation is

$$\hat{e}_{m,j} = \widehat{U}_{m,j} - \hat{u}_m(x_{m,j}, s), \tag{A.6}$$

where $\hat{u}_m(x,s)$ is the Laplace transform of $u_m(x,t)$. The error satisfies the equations

$$s\hat{e}_{1,j} = \mathcal{D}_1 D_+ D_- \hat{e}_{1,j} + \mathcal{D}_1 \frac{h_1^2}{12} \partial_x^4 \hat{u}_1(\xi_{1,j}, s), \quad \text{for } j = 0,-1,-2,\ldots, \tag{A.7}$$

$$s\hat{e}_{2,j} = \mathcal{D}_2 D_+ D_- \hat{e}_{2,j} + \mathcal{D}_2 \frac{h_1^2}{12} \partial_x^4 \hat{u}_2(\xi_{2,j}, s), \quad \text{for } j = 0,1,2,\ldots, \tag{A.8}$$

$$\mathcal{K}_1 D_0 \hat{e}_{1,0} = \mathcal{K}_2 D_0 \hat{e}_{2,0} - \mathcal{K}_1 \frac{h_1^2}{6} \partial_x^3 \hat{u}_1(\eta_{1,0}, s) - \mathcal{K}_2 \frac{h_2^2}{6} \partial_x^3 \hat{u}_2(\eta_{2,0}, s), \tag{A.9}$$

$$\hat{e}_{1,0} = \hat{e}_{2,0}, \tag{A.10}$$

$$\|\hat{e}_1\|_h < \infty, \quad \|\hat{e}_2\|_h < \infty, \tag{A.11}$$

for some $\xi_{m,j} \in [x_{m,j-1}, x_{m,j+1}]$, and $\eta_{m,j} \in [x_{m,j-1}, x_{m,j+1}]$. In the usual way we can subtract out functions that make the forcing terms in the first two equations zero and that only changes the forcings in the boundary conditions by $O(h_1^2 + h_2^2)$. This results in a new error equation with inhomogeneous boundary conditions for the error functions $W_{1,j}$ and $W_{2,j}$,

$$sW_{1,j} = \mathcal{D}_1 D_+ D_- W_{1,j}, \quad \text{for } j = 0,-1,-2,\ldots, \tag{A.12}$$

$$sW_{2,j} = \mathcal{D}_2 D_+ D_- W_{2,j}, \quad \text{for } j = 0,1,2,\ldots, \tag{A.13}$$

$$\mathcal{K}_1 D_0 W_{1,0} = \mathcal{K}_2 D_0 W_{2,0} + C_1 h^2, \tag{A.14}$$

$$W_{1,0} = W_{2,0} + C_2 h^2. \tag{A.15}$$

$$\|W_1\|_h < \infty, \quad \|W_2\|_h < \infty, \tag{A.16}$$

where C_1 and C_2 are constants and h^2 denotes a term of order $O(h_1^2 + h_2^2)$. The solution to the problem (A.12)–(A.16) is of the form

$$W_{1,j} = A\kappa_1^{-j}, \quad W_{2,j} = B\kappa_2^j,$$

where

$$\kappa_m = 1 + s_m/2 - \sqrt{s_m + s_m^2}$$

is a root of the characteristic equation

$$\kappa - 2 + \kappa^{-1} = s_m$$

with $s_m = sh_m^2/\mathcal{D}_m$, for $m = 1, 2$. To satisfy (A.16), the branch of the square root is taken so that $|\kappa_m| < 1$ for $\mathrm{Re}(s) > 0$. Applying the interface conditions (A.14) and (A.15) implies

$$A\mathcal{K}_1\frac{\kappa_1^{-1}-\kappa_1}{2h_1}=B\mathcal{K}_2\frac{\kappa_2-\kappa_2^{-1}}{2h_2}+C_1h^2,$$
$$A=B+C_2h^2$$

with solution

$$A=D^{-1}\left(C_1h^2-C_2h^2\mathcal{K}_2\frac{\kappa_2-\kappa_2^{-1}}{2h_2}\right), \tag{A.17}$$
$$B=A-C_2h^2, \tag{A.18}$$
$$D=\frac{\mathcal{K}_1}{h_1}(s_1/2+1-\kappa_1)+\frac{\mathcal{K}_2}{h_2}(s_2/2+1-\kappa_2). \tag{A.19}$$

To show the accuracy of the scheme we consider the limit $sh_m^2\ll 1$, which implies

$$\kappa_m\sim 1-\sqrt{s/\mathcal{D}_m}h_m+\frac{1}{2}sh_m^2/\mathcal{D}_m+O(h_m^3), \tag{A.20}$$
$$\frac{\kappa_m-\kappa_m^{-1}}{2h_m}\sim-\sqrt{s/\mathcal{D}_m}+O(h_m^2), \tag{A.21}$$
$$D\sim\mathcal{K}_1\sqrt{s/\mathcal{D}_1}+\mathcal{K}_2\sqrt{s/\mathcal{D}_2}+O(h^2), \tag{A.22}$$
$$A\sim\left[\mathcal{K}_1\sqrt{s/\mathcal{D}_1}+\mathcal{K}_2\sqrt{s/\mathcal{D}_2}\right]^{-1}\left(C_1h^2+C_2h^2\mathcal{K}_2\sqrt{s/\mathcal{D}_2}\right). \tag{A.23}$$

The scheme is therefore second-order accurate since $A=O(h^2)$ and $B=O(h^2)$ for $sh_m^2\ll 1$. For stability we need to prove that D is bounded away from zero for $\mathrm{Re}(s)>0$. Let $\kappa_m=\alpha_m+i\beta_m$ be the complex representation for κ_m, where α_m and β_m are real. Whence from (A.19),

$$\mathrm{Re}(D)=\frac{\mathcal{K}_1}{h_1}(\mathrm{Re}(s_1)/2+1-\alpha_1)+\frac{\mathcal{K}_2}{h_2}(\mathrm{Re}(s_2)/2+1-\alpha_2), \tag{A.24}$$
$$=\frac{1}{2}(\mathcal{K}_1h_1/\mathcal{D}_1+\mathcal{K}_2h_2/\mathcal{D}_2)\mathrm{Re}(s)+\frac{\mathcal{K}_1}{h_1}(1-\alpha_1)+\frac{\mathcal{K}_2}{h_2}(1-\alpha_2). \tag{A.25}$$

Note that $|\kappa_m|<1$ implies $|\alpha_m|<1$ and $1-\alpha_m>0$. Therefore,

$$\mathrm{Re}(D)>\frac{1}{2}(\mathcal{K}_1h_1/\mathcal{D}_1+\mathcal{K}_2h_2/\mathcal{D}_2)\mathrm{Re}(s)>0,\quad \text{for } \mathrm{Re}(s)>0,$$

and the scheme is stable. This completes the proof of the theorem. □

Theorem 2. *Solutions to Eqs. (44)–(50) are stable in the sense of Godunov–Ryabenkii provided the time-step Δt satisfies the von Neumann stability conditions*

$$d_m\equiv\frac{\mathcal{D}_m\Delta t}{h_m^2}<\frac{1}{2},\quad m=1,2. \tag{A.26}$$

Proof. Proceeding along the lines of the previous proof, the solution to the homogeneous equations will be of the form

$$U_{1,j}^n=z^n\kappa_1^{-j},\quad j=1,0,-1,-2,\ldots, \tag{A.27}$$
$$U_{2,j}^n=z^n\kappa_2^{j},\quad j=-1,0,1,2,3,\ldots, \tag{A.28}$$

where

$$\kappa_m=1+s_m/2\pm\sqrt{s_m+s_m^2/4}, \tag{A.29}$$
$$=1+\frac{z-1}{2d_m}\pm\sqrt{\frac{z-1}{d_m}\left(1+\frac{z-1}{4d_m}\right)} \tag{A.30}$$

with $s_m=(z-1)/d_m$ and where we choose the branch of the square root so that $|\kappa_m|<1$ for $|z|>1$. Substitution of the expressions for κ_m into the interface condition (46) and using $\kappa_m-\kappa_m^{-1}=2(\kappa_m-1)-s_m$, gives a nonlinear equation satisfied by z,

$$(\pm)\frac{\mathcal{K}_1}{h_1}\sqrt{\frac{z-1}{d_1}\left(1+\frac{z-1}{4d_1}\right)}=-(\pm)\frac{\mathcal{K}_2}{h_2}\sqrt{\frac{z-1}{d_2}\left(1+\frac{z-1}{4d_2}\right)}. \tag{A.31}$$

The sign taken for each square root will depend on z and d_m and we note the possibilities with $(\pm)$. We need to show that there are no solutions to this equation with $|z| > 1$. We first show that all solutions to (A.31) are real. Taking the square of (A.31) implies that

$$\frac{\mathcal{K}_1^2}{h_1^2}\frac{z-1}{d_1}\left(1+\frac{z-1}{4d_1}\right)=\frac{\mathcal{K}_2^2}{h_2^2}\frac{z-1}{d_2}\left(1+\frac{z-1}{4d_2}\right). \tag{A.32}$$

Thus $z = 1$ or z satisfies

$$\alpha(z-1)=\beta-1, \quad \text{with } \alpha=\frac{1}{4d_1}-\frac{\beta}{4d_2}, \text{ and } \beta=\frac{\mathcal{K}_2^2}{h_2^2}\frac{h_1^2}{\mathcal{K}_1^2}. \tag{A.33}$$

If $\alpha \neq 0$ then $z = 1 + (\beta - 1)/\alpha$ is another real root. The special case $\alpha = 0$ and $\beta = 1$ suggests that any value of z could be a solution. However this is a spurious solution introduced when squaring expression (A.31) since in this case $d_1 = d_2,\ s_1 = s_2$ and $\kappa_1 = \kappa_2$ (implying that both square roots take the same sign) and (A.31) becomes

$$\sqrt{1+\frac{z-1}{4d_1}}=-\sqrt{1+\frac{z-1}{4d_1}} \tag{A.34}$$

with $z = 1 - 4d_1,\ z \in \mathbb{R}$, as the only solution. Therefore, given that solutions for z are real and $|z| > 1$, there are two cases to consider, $z > 1$ and $z < -1$. Define $D(z)$ by

$$D(z)=(\pm)\frac{\mathcal{K}_1}{h_1}\sqrt{\frac{z-1}{d_1}\left(1+\frac{z-1}{4d_1}\right)}+(\pm)\frac{\mathcal{K}_2}{h_2}\sqrt{\frac{z-1}{d_2}\left(1+\frac{z-1}{4d_2}\right)}. \tag{A.35}$$

We need to show that there are no solutions to $D(z) = 0$ when $z > 1$ or when $z < -1$. If $z > 1$ then from (A.30) we note that the arguments of the square roots are positive and also that the negative sign for the square roots must be taken to make $|\kappa_m| < 1$. This implies that $D(z)$ is the sum of two negative quantities and thus there can be no real solutions for $z > 1$. When $z < -1$, and $d_m < \frac{1}{2}$ then the arguments of the square roots are also positive but we must take the positive sign for both square roots to make $|\kappa_m| < 1$. Thus in this case $D(z)$ is the sum of two positive quantities and therefore there are no real roots with $z < -1$. We have thus shown that there are no solutions with $|z| > 1$, proving the theorem. □

References

[1] M.B. Giles, Stability analysis of numerical interface conditions in fluid–structure thermal analysis, Int. J. Numer. Method Eng. 25 (1997) 421–436.
[2] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1) (1990) 1–64.
[3] G. Chesshire, W. Henshaw, A scheme for conservative interpolation on overlapping grids, SIAM J. Sci. Comput. 15 (4) (1994) 819–845.
[4] W.D. Henshaw, D.W. Schwendeman, An adaptive numerical scheme for high-speed reactive flow on overlapping grids, J. Comput. Phys. 191 (2003) 420–447.
[5] W.D. Henshaw, D.W. Schwendeman, Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow, J. Comput. Phys. 216 (2) (2006) 744–779.
[6] E.A. Volkov, A finite difference method for finite and infinite regions with piecewise smooth boundaries, Doklady 168 (5) (1966) 744–747.
[7] E.A. Volkov, The method of composite meshes for finite and infinite regions with piecewise smooth boundaries, Proc. Steklov Inst. Math. 96 (1968) 145–185.
[8] G. Starius, Composite mesh difference methods for elliptic and boundary value problems, Numer. Math. 28 (1977) 243–258.
[9] G. Starius, On composite mesh difference methods for hyperbolic differential equations, Numer. Math. 35 (1980) 241–255.
[10] G. Starius, Constructing orthogonal curvilinear meshes by solving initial value problems, Numer. Math. 28 (1977) 25–48.
[11] B. Kreiss, Construction of a curvilinear grid, SIAM J. Sci. Stat. Comput. 4 (2) (1983) 270–279.
[12] J.L. Steger, J.A. Benek, On the use of composite grid schemes in computational aerodynamics, Comput. Method Appl. Mech. Eng. 64 (1987) 301–320.
[13] A. Kapila, D. Schwendeman, J. Bdzil, W. Henshaw, A study of detonation diffraction in the ignition-and-growth model, Combust. Theor. Model. 11 (5) (2007) 781–822.
[14] W.D. Henshaw, D.W. Schwendeman, Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement, J. Comput. Phys. 227 (16) (2008) 7469–7502.
[15] J.W. Banks, D.W. Schwendeman, A.K. Kapila, W.D. Henshaw, A high-resolution Godunov method for compressible multi-material flow on overlapping grids, J. Comput. Phys. 223 (2007) 262–297.
[16] J.W. Banks, D.W. Schwendeman, A.K. Kapila, W.D. Henshaw, A study of detonation propagation and diffraction with compliant confinement, Combust. Theor. Model. 12 (4) (2008) 769–808.
[17] J.Y. Tu, L. Fuchs, Calculation of flows using three-dimensional overlapping grids and multigrid methods, Int. J. Numer. Method Eng. 38 (1995) 259–282.
[18] P.G. Buning, I.T. Chiu, S. Obayashi, Y.M. Rizk, J.L. Steger, Numerical simulation of the integrated space shuttle vehicle in ascent, Paper 88-4359-CP, AIAA, 1988.
[19] M. Hinatsu, J. Ferziger, Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique, Int. J. Numer. Method Fluid 13 (1991) 971–997.
[20] R. Meakin, Moving body overset grid methods for complete aircraft tiltrotor simulations, Paper 93-3350, AIAA, 1993.
[21] D. Pearce, S. Stanley, F. Martin, R. Gomez, G.L. Beau, P. Buning, W. Chan, T. Chui, A. Wulf, V. Akdag, Development of a large scale Chimera grid system for the space shuttle launch vehicle, Paper 93-0533, AIAA, 1993.
[22] R. Maple, D. Belk, A new approach to domain decomposition, the Beggar code, in: N. Weatherill (Ed.), Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Pineridge Press Limited, 1994, pp. 305–314.
[23] D. Jespersen, T. Pulliam, P. Buning, Recent enhancements to OVERFLOW, Paper 97-0644, AIAA, 1997.
[24] R.L. Meakin, Composite overset structured grids, in: J.F. Thompson, B.K. Soni, N.P. Weatherill (Eds.), Handbook of Grid Generation, CRC Press, 1999, pp. 1–20 (Ch. 11).

[25] C. Kiris, D. Kwak, S. Rogers, I. Chang, Computational approach for probing the flow through artificial heart devices, J. Biomech. Eng. 119 (4) (1997) 452–460.
[26] W.D. Henshaw, A high-order accurate parallel solver for Maxwell's equations on overlapping grids, SIAM J. Sci. Comput. 28 (5) (2006) 1730–1765. <http://link.aip.org/link/?SCE/28/1730/1>.
[27] Y. Tahara, R. Wilson, P. Carrica, F. Stern, RANS simulation of a container ship using a single-phase level-set method with overset grids and the prognosis for extension to a self-propulsion simulator, J. Mar. Sci. Technol. 11 (4) (2006) 209–228.
[28] F. Olsson, J. Yström, Some properties of the upper convected Maxwell model for viscoelastic fluid flow, J. Non-Newtonian Fluid Mech. 48 (1993) 125–145.
[29] N.A. Petersson, A numerical method to calculate the two-dimensional flow around an underwater obstacle, SIAM J. Numer. Anal. 29 (1992) 20–31.
[30] P. Fast, Dynamics of interfaces in non-Newtonian Hele–Shaw flow, Ph.D. Thesis, New York University, Courant Institute of Mathematical Sciences, 1999.
[31] P. Fast, M.J. Shelley, A moving overset grid method for interface dynamics applied to non-Newtonian Hele–Shaw flow, J. Comput. Phys. 195 (2004) 117–142.
[32] K.-H. Kao, M.-S. Liou, Application of Chimera/unstructured hybrid grids to conjugate heat transfer, AIAA J. 35 (9) (1997) 1472–1478.
[33] S. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, 1980.
[34] G. Desrayaud, A. Fichera, G. Lauriat, Natural convection air-cooling of a substrate-mounted protruding heat source in a stack of parallel boards, Int. J. Heat Fluid Flow 28 (2007) 469–482.
[35] M. Schäfer, I. Teschauer, Numerical simulation of coupled fluid–solid problems, Comput. Method Appl. Mech. Eng. 190 (2001) 3645–3667.
[36] N. Wansophark, A. Malatip, P. Dechaumphai, Streamline upwind finite element method for conjugate heat transfer problems, Acta Mech. Sin. 21 (2005) 436–443.
[37] X. Chen, P. Han, A note on the solution of conjugate heat transfer problems using SIMPLE-like algorithms, Int. J. Heat Fluid Flow 21 (2000) 463–467.
[38] A. Liaqat, A. Baytas, Numerical comparison of conjugate and non-conjugate natural convection for internally heated semi-circular pools, Int. J. Heat Fluid Flow 22 (2001) 650–656.
[39] N.B. Sambamurthy, A. Shaija, G.S.V.L. Narasimham, M.V. Krishna Murthy, Laminar conjugate natural convection in horizontal annuli, Int. J. Heat Fluid Flow 29 (5) (2008) 1347–1359.
[40] B. Roe, R. Jaiman, A. Haselbacher, P.H. Geubelle, Combined interface boundary condition method for coupled thermal simulations, Int. J. Numer. Method Fluid 57 (2008) 329–354.
[41] D. Funaro, A. Quarteroni, P. Zanolli, An iterative procedure with interface relaxation for domain decomposition methods, SIAM J. Numer. Anal. 25 (6) (1998) 1213–11236.
[42] J.R. Rice, P. Tsompanopoulou, E. Vavalis, Fine tuning interface relaxation methods for elliptic differential equations, Appl. Numer. Math. 43 (4) (2002) 459–481.
[43] M. Garbey, Acceleration of the Schwarz method for elliptic problems, SIAM J. Sci. Comput. 26 (6) (2005) 1871–1893.
[44] W. Henshaw, A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids, J. Comput. Phys. 113 (1) (1994) 13–25.
[45] W. Henshaw, H.-O. Kreiss, L. Reyna, A fourth-order accurate difference approximation for the incompressible Navier–Stokes equations, Comput. Fluid 23 (4) (1994) 575–593.
[46] W.D. Henshaw, N.A. Petersson, A split-step scheme for the incompressible Navier–Stokes equations, in: M. Hafez (Ed.), Numerical Simulation of Incompressible Flows, World Scientific, 2003, pp. 108–125.
[47] W.D. Henshaw, On multigrid for overlapping grids, SIAM J. Sci. Comput. 26 (5) (2005) 1547–1572.
[48] W. Henshaw, Ogen: an overlapping grid generator for Overture, Research Report UCRL-MA-132237, Lawrence Livermore National Laboratory, 1998.
[49] N.A. Petersson, Stability of pressure boundary conditions for Stokes and Navier–Stokes equations, J. Comput. Phys. 172 (1) (2001) 40–70.
[50] B. Gustafsson, H.-O. Kreiss, J. Oliger, Time Dependent Problems and Difference Methods, John Wiley and Sons Inc., 1995.
[51] P. Roache, Code verification by the method of manufactured solutions, ASME J. Fluid Eng. 124 (1) (2002) 4–10.
[52] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, The portable extensible toolkit for scientific computation, Technical Report, Argonne National Laboratory, 1999. <http://www.mcs.anl.gov/petsc/petsc.html>.
[53] J.J. Sanchez, W.H. Giedt, Thin films for reducing tamping gas convection heat transfer effects in a National Ignition Facility hohlraum, Fus. Sci. Technol. 44 (2003) 811–819.
[54] R.A. London, R.L. McEachern, B.J. Kozioziemski, D.N. Bittner, Computational design of infrared enhanced layering of ICF capsules, Fus. Sci. Technol. 45 (2) (2004) 245–252.
[55] G. Collins, P. Souers, E. Fearon, E. Mapoles, R. Tsugawa, J. Gaines, Thermal conductivity of condensed $D - T$ and T_2, Phys. Rev. B 41 (4) (1990) 1816–1823.
[56] J. Lamarsch, A. Baratta, Introduction to Nuclear Engineering, third ed., Prentice Hall, New Jersey, 2001.
[57] W. Henshaw, Mappings for Overture, a description of the Mapping class and documentation for many useful Mappings, Research Report UCRL-MA-132239, Lawrence Livermore National Laboratory, 1998.