

CPC 50th anniversary article

Semtex: A spectral element–Fourier solver for the incompressible Navier–Stokes equations in cylindrical or Cartesian coordinates

H.M. Blackburn^{*}, D. Lee, T. Albrecht, J. Singh

Department of Mechanical and Aerospace Engineering, Monash University, Vic 3800, Australia

ARTICLE INFO

Article history:

Received 25 March 2019
 Received in revised form 24 May 2019
 Accepted 27 May 2019
 Available online 30 July 2019

Keywords:

Navier–Stokes
 Spectral elements
 Cylindrical coordinates
 Direct numerical simulation

ABSTRACT

Semtex enables direct numerical simulation (DNS) of the incompressible Navier–Stokes equations by coupling continuous-Galerkin nodal spectral element–Fourier spatial discretisation with semi-implicit temporal integration via a time-splitting scheme. Transport of a scalar quantity may be included. The analyst has a choice of Cartesian or cylindrical coordinate systems. Domain geometries and solutions may be two-dimensional with spectral element decomposition of arbitrary planar shapes, or made three-dimensional by extrusion along a spatially homogeneous direction in which Fourier expansions are employed. For three-dimensional solutions, MPI may be used to support parallel execution. Various body forces, including Boussinesq buoyancy and Coriolis terms may be added to the momentum equation to simulate e.g. the effects of stratification and thermal expansion or reference frame rotation. Parallel decomposition is performed in the Fourier dimension only, and two-dimensional elliptic systems in the plane are solved for the spectral element discretisation using direct (Cholesky) or iterative (conjugate-gradient) methods. *Semtex* includes a suite of additional tools for generating initial conditions and model configurations, for post processing and for analysis of model output.

Program summary*Program Title:* *Semtex**Program Files doi:* <http://dx.doi.org/10.17632/65mz2szz5t.1>*Code Ocean Capsule:* <https://doi.org/10.24433/CO.2589809.v1>*Licensing provisions:* GPLv2*Programming languages:* C++, C, Fortran 77*External routines:* BLAS, LAPACK, *yacc*/ *bison*, (optionally) MPI*Nature of problem:* Two- or three-dimensional incompressible Navier–Stokes in cylindrical and periodic Cartesian geometries with optional body forces. Two- or three-component velocity fields.*Solution method:* Continuous Galerkin nodal spectral element–Fourier spatial discretisation with semi-implicit time-splitting-based temporal integration of the nonlinear, viscous and pressure gradient terms in the Navier–Stokes equations via a projection method.

Crown Copyright © 2019 Published by Elsevier B.V. All rights reserved.

1. Introduction

Following its inception in the mid-1980s [1] the spectral element method has proven to be a highly popular approach for the modelling of incompressible and low-Mach-number flows within a variety of engineering and geophysical problems. These methods balance the exponential ('spectral') convergence of errors associated with global collocation methods (e.g. Fourier or Chebyshev pseudospectral methods) with the geometric flexibility of traditional low-order finite element methods. This is typically achieved via the use of high-order tensor-product polynomials with compact support within each element and Jacobi-

polynomial-based quadrature rules both for defining and integrating these polynomials [2–4]. The spectral element method resulted from embedding these high-order approaches within the framework of more traditional finite element techniques [e.g. 5]; a key idea is that spectral element methods *are* finite element methods.

The spectral element–Fourier spatial discretisation is well-suited to the solution of incompressible flow problems in cylindrical and Cartesian geometries in which at least one homogeneous direction exists. Variations of the spectral element methodology for Cartesian coordinates which incorporated one-dimensional Fourier expansions appeared soon after the original description of the spectral element method [6–9]. Subsequently this treatment was elaborated to include cylindrical coordinates, with demonstration of full spectral convergence in all directions [10].

^{*} Corresponding author.

E-mail address: hugh.blackburn@monash.edu (H.M. Blackburn).

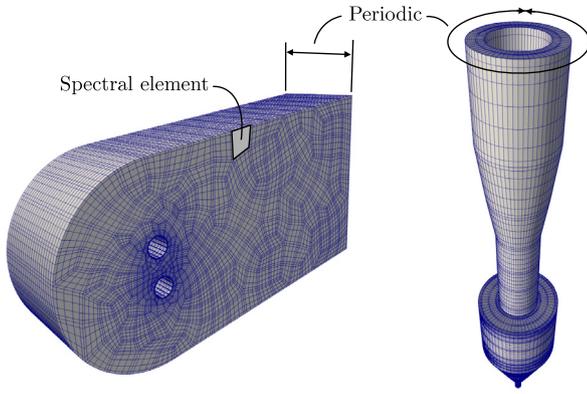


Fig. 1. Example three-dimensional spectral element–Fourier meshes in Cartesian (left) and cylindrical (right) geometries as used in *Semtex*. Parallel domain decomposition is applied in the Fourier (periodic) dimension only.

A velocity-correction projection method is used for solving for the elliptic pressure and viscous terms at intermediate steps [11–13], while parallel domain decomposition is applied across the Fourier dimension only, so that the spectral element elliptic operators are solved without the need for parallel communications.

Principal features of *Semtex*:

- Cartesian or cylindrical coordinate formulations;
- Edge-conforming quadrilateral nodal tensor-product Gauss–Lobatto–Legendre-based two-dimensional spectral element shape functions in the (x, y) plane;
- Fourier expansions in z coordinate as required;
- DNS of the incompressible Navier–Stokes equations with (spatial domain/number of components): two-dimensional/two-component (2D2C), two-dimensional/three-component (2D3C), three-dimensional/three-component (3D3C) velocity fields and first-, second- or third-order fractional-step time integration;
- Optional MPI-based parallel solutions of three-dimensional problems, made concurrent across two-dimensional Fourier modes;
- Optional coupled solution of scalar advection–diffusion equation;
- A range of user-definable forcing terms for the Navier–Stokes equations including rotating reference frame accelerations and Boussinesq buoyancy;
- Fast direct solver for continuous–Galerkin treatment of two-dimensional elliptic equations;
- Pseudospectral, non-dealiased evaluation of nonlinear product terms with skew-symmetric form as default;
- A built-in function parser based on *yacc/bison* for evaluation of user-defined constants, initial conditions, boundary conditions and forcing terms;
- A range of adjunct utility programmes for pre- and post-processing;
- An extensive user guide.
- Automated compilation system based on *cmake*.

The remainder of this manuscript details the formulation and use of *Semtex* for incompressible Navier–Stokes flows in a variety of geometric and physical configurations. Section 2 briefly discusses the spectral element–Fourier discretisation of *Semtex*, as well as the formulation of the solver. For more detailed discussion the reader is referred to standard texts on the underlying methods [2–5]. Section 3 introduces the specific capabilities of

Semtex including various non-standard forcing terms, boundary conditions, geometric specifications and additional stand alone utilities for pre- and post-processing of data files. In Section 4 we present some example applications of *Semtex* for the purposes of validating against previously published results and exploring the capabilities of the code to model more exotic flow regimes. Finally a brief review of the capabilities and limitations of *Semtex* will be presented in Section 5.

In Sections 3 and 4, descriptions will be given as to how to set various model configuration parameters within *Semtex*. Generally speaking, these are specified within an ASCII ‘session’ input file for a given model run. A session file includes all relevant information for a given model configuration, including solver parameters, model geometry, boundary and initial conditions. A full description of session file information is included in the *Semtex* user guide, which is supplied with the code distribution.

2. Discretisation

The foundation of the spectral element method as implemented in *Semtex* is the discretisation of the two dimensional planar domain $\Omega \in \mathbb{R}^2$ into a set of contiguous edge-conforming quadrilateral elements Ω^e , and the use of nodal Legendre cardinal functions (Lagrange interpolants) within each element in order to construct a polynomial function space with C^0 continuity across element boundaries [2–4].

2.1. Element-level operations

The Legendre cardinal functions for a polynomial of degree N are defined on the standard one-dimensional region $r \in [-1, +1]$ as

$$h_i(r) = \frac{1}{N(N+1)} \frac{(1-r^2) L'_N(r)}{(r_i-r) L'_N(r_i)}, \quad (1)$$

where r_i , $0 \leq i \leq N$ are the Gauss–Lobatto–Legendre (GLL) quadrature nodes within the canonical domain, $L_N(r)$ is the N^{th} degree Legendre polynomial and $L'_N(r)$ its derivative. The set of Lagrange interpolants and associated GLL nodes for $N = 6$ are shown in Fig. 2. Such Lagrange basis functions, each of which is unity at one GLL node and zero at the others, are commonly referred to as nodal basis functions [3, §2.3.4.2]. The properties of Legendre polynomials are documented in [14]; related quadratures and derivatives in Appendix B of [2]; computational algorithms for Gauss–Lobatto–Legendre quadrature nodes and weights (often referred to as the zeros z_i and weights w_i of the quadrature scheme) are discussed in Appendices B and C of [3].

While the basis functions are not themselves Legendre polynomials, they share the asymptotic approximation characteristics of those functions owing to the choice of GLL quadrature nodes [15, §5.4.3]. We note that if one wishes to integrate the Lagrange interpolants (1) over the standard region one has the choice of using the same number of GLL nodes (and quadrature weights) as used to define the polynomials, or more (to increase the accuracy of the estimate), or even to use the standard Gauss points. *Semtex* uses simple ‘equal-order’ GLL quadrature, with the quadrature points at the nodes of the shape functions as indicated in Fig. 2. This choice has a number of convenient outcomes which stem from the fact that $h_i(r_j) \equiv \delta_{ij}$ where δ_{ij} is the Kronecker delta function.

From the one-dimensional basis functions (1) we next move to constructing two-dimensional elements and examine local elemental operations. The representation of functions in each quadrilateral element is spanned by a set of two-dimensional basis functions constructed as tensor-product combinations of equal-order Lagrange interpolants such that

$$\psi_{ij}(r, s) = h_i(r)h_j(s), \quad 0 \leq i, j \leq N, \quad (2)$$

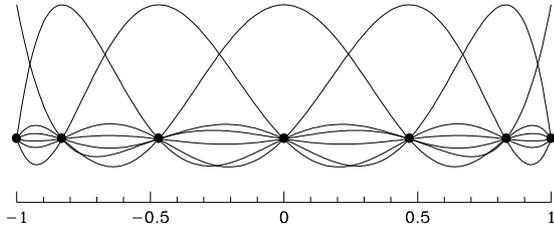


Fig. 2. Legendre polynomial cardinal functions and Gauss-Lobatto points for polynomial degree $N = 6$ (note that the number of points is $N + 1$).

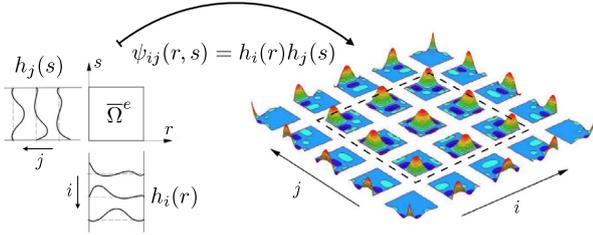


Fig. 3. Creation of two-dimensional elemental basis functions as tensor products of one-dimensional basis functions. Note that these functions can be partitioned into those with only interior support (shown within dashed line) and those with some exterior support.

where the canonical or standard region of the element is given as $\bar{\Omega}^e = [-1, +1] \times [-1, +1]$, see Fig. 3. A significant feature of these basis functions (which carries over from the one-dimensional functions of Fig. 2) is that they can be partitioned into two sets: those which are zero all around the exterior boundary (i.e. with only interior support) and those which are non-zero on at least one element edge (i.e. with partial exterior support) [see3, § 3.1.1.1].

In what follows, pairs of integers such as i, j or p, q may be taken as row and column indices within rectangular elements. Sometimes for convenience, we need a single index over the entire set of unknowns in an element, e.g. $k = jN + i$.

A standard finite element approach which allows non-rectangular quadrilateral elements to be obtained, and with that, the possibility of unstructured meshes, is the adoption of isoparametric mapping of element shapes and shape functions [5, Ch. 3]. To achieve this, the (x, y) positions of the external (r, s) nodes of each element are first computed based on a proportional mapping of arc length along the appropriate side in (x, y) space. In *Semtex* these sides may be defined as straight lines, circular arcs, or splined curves. Subsequently the (x, y) positions of internal nodes are generated via bilinear interpolation (a Coons patch). Interpolation of geometric positions and basis functions within an element in (x, y) space is carried out using the basis functions defined on the master (r, s) domain. This idea is demonstrated in Fig. 4. Formally, local coordinates $\mathbf{r} = (r, s)$ within each element e may be transformed to global coordinates $\mathbf{x}^e = (x^e, y^e)$ via the isoparametric mapping using the discrete elemental nodal coordinates \mathbf{x}_{ij}^e and continuous basis functions

$$\mathbf{x}^e(r, s) \approx \sum_{i=0}^N \sum_{j=0}^N \mathbf{x}_{ij}^e h_i(r) h_j(s). \quad (3)$$

From this it follows that one may approximate spatial partial derivatives within elements as e.g.

$$\frac{\partial \mathbf{x}^e}{\partial r} \approx \sum_{i=0}^N \sum_{j=0}^N \mathbf{x}_{ij}^e \frac{dh_i(r)}{dr} h_j(s), \quad (4)$$

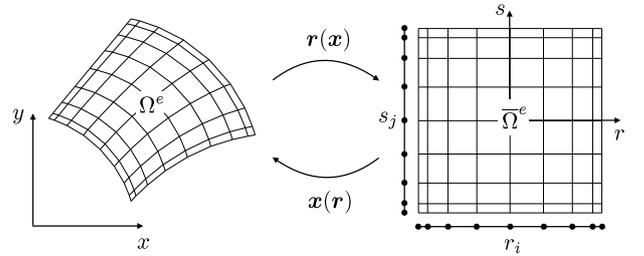


Fig. 4. The idea of (isoparametric) mapping between local elemental physical (x^e, y^e) space and master (r, s) space.

$$\frac{\partial \mathbf{x}^e}{\partial s} \approx \sum_{i=0}^N \sum_{j=0}^N \mathbf{x}_{ij}^e h_i(r) \frac{dh_j(s)}{ds}. \quad (5)$$

In the following we refer to these partial derivatives of physical global coordinates with respect to standard coordinates as ‘inverse partials’.

Likewise, we can continuously approximate any variable within an element using the basis functions; e.g. for the field variable u defined at the nodes i, j

$$u^e(r, s) \approx \sum_{i=0}^N \sum_{j=0}^N u_{ij}^e h_i(r) h_j(s) \quad (6)$$

and its partial derivatives, e.g.

$$\frac{\partial u^e}{\partial r} = \sum_{i=0}^N \sum_{j=0}^N u_{ij} \frac{dh_i(r)}{dr} h_j(s). \quad (7)$$

We are usually principally interested in approximating these partial derivatives at element nodal points, say r_p and s_q . The derivatives at the nodal points can be interpreted as a two-dimensional derivative operator matrix D

$$D_{pi} = \frac{dh_i(r_p)}{dr}, \quad (8)$$

whose values are readily found using the properties of Lagrange interpolants [e.g. 14, 25.3.2]. Evaluation of partial derivatives with respect to master element coordinates (r, s) at nodal points, such as those in (4) and (5) is then (considering the nodal values of variables within each element to be stored as a two-dimensional array) efficiently accomplished via matrix-matrix pre- or post-multiplication with either the matrix D or its transpose D^T using e.g. BLAS routine `dgemm`.

However, one often requires partial derivatives of variables with respect to the physical coordinates (x, y) , for which the chain rule is invoked, e.g. $\partial u / \partial y = \partial u / \partial r \times \partial r / \partial y + \partial u / \partial s \times \partial s / \partial y$. For these one must construct ‘forward partials’ such as $\partial r / \partial y$. Since (again via chain rule)

$$d\mathbf{x} = \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} dr \\ ds \end{bmatrix} = J d\mathbf{r}, \quad (9)$$

$d\mathbf{r} = J^{-1} d\mathbf{x} = |J|^{-1} \text{adj}(J) d\mathbf{x}$ where $|J| = \det(J) = \partial x / \partial r \times \partial y / \partial s - \partial x / \partial s \times \partial y / \partial r$ is the determinant of the Jacobian matrix J and $\text{adj}(J)$ its adjunct. The partial derivatives in (9) can be approximated at any point in an element using e.g. (4). But since also (chain rule once more)

$$d\mathbf{r} = \begin{bmatrix} dr \\ ds \end{bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial x} & \frac{\partial s}{\partial y} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}, \quad (10)$$

forward partial terms may be found using the definition of the adjunct and by equating [3, §4.1.3.4]

$$\begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial s}{\partial s} & \frac{\partial s}{\partial y} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} \frac{\partial y}{\partial s} & -\frac{\partial x}{\partial s} \\ -\frac{\partial y}{\partial r} & \frac{\partial x}{\partial r} \end{bmatrix}. \quad (11)$$

We now have the means to compute both forward and inverse partial derivatives at nodal locations within each element, and with that, the ability to approximate terms of PDEs cast in physical spatial coordinates.

Next, consider the task of finding the unit outward normal vector \mathbf{n} along the edge of an element, say one for which $r = r(x, y) = \text{const}$, i.e. the edge is a contour line of r . Then the direction of \mathbf{n} is readily found as $\nabla r = (\partial r/\partial x, \partial r/\partial y)$.

The integral of a variable over an elemental domain Ω^e is approximated using GLL quadrature and the Jacobian of the mapping between Ω^e and $\overline{\Omega}^e$, e.g.

$$\begin{aligned} \int_{\Omega^e} u(x, y) dx dy &= \int_{\overline{\Omega}^e} u(r, s) |J|(r, s) dr ds \\ &\approx \sum_{i=0}^N \sum_{j=0}^N u_{ij} |J|_i w_i w_j, \end{aligned} \quad (12)$$

where w_i, w_j are quadrature weights. Similar considerations apply to approximation of the integral of a variable along the edge of an element.

The approximation (12) is sometimes interpreted as the contraction of the product of the elemental unknowns u_k (considered as a single-index vector) with its ‘mass matrix’ M as $\sum_{k=0}^N M_{kk} u_k = |J|_i w_i w_j u_{ij}$ where, conveniently, for equal-order GLL quadrature and the nodal basis, M is a diagonal matrix.

2.2. Temporal discretisation of the Navier–Stokes equations for DNS

To help establish motivation for the treatment of global operations with spectral element approximations, consider the incompressible Navier–Stokes equations for the velocity field $\mathbf{u}(\mathbf{x}, t)$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad (13)$$

where, for a fluid of constant density ρ_0 , $P \equiv p/\rho_0$ and $\nu = \mu/\rho_0$ is the constant kinematic viscosity of the fluid whose (dynamic) viscosity is μ , while $\mathbf{f}(\mathbf{x}, t)$ represents body force per unit mass. In what follows, the nonlinear terms are sometimes given the brief notation $\mathbf{N}(\mathbf{u}) = \mathbf{u} \cdot \nabla \mathbf{u}$; we note also the continuum-variable equivalence (which relies on the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$) $\mathbf{N}(\mathbf{u}) = \nabla \cdot (\mathbf{u}\mathbf{u})$, where $\mathbf{u}\mathbf{u}$ is a dyadic. Starting from a given initial condition $\mathbf{u}(\mathbf{x}, 0)$ the equation set (13) must be integrated forwards in time.

Temporal integration in *Semtex* is handled using a ‘stiffly-stable’ [16] approximation for the derivative of scalar variable u at time level $(n+1)$, based on backwards differencing in time

$$\partial_t u^{(n+1)} \approx (\Delta t)^{-1} \sum_{q=0}^K \alpha_q u^{(n+1-q)} \quad (14)$$

where Δt is a constant time step and α_q are a set of weights. For $K = 1$, the method is the backwards (or implicit) Euler approximation with $\alpha_0 = 1$ and $\alpha_1 = -1$. The approximation (14) has an error $O(\Delta t)^{K+1}$; *Semtex* can be run with $K = 1, 2$, or 3 , producing successively smaller errors but carrying the penalties of reduction of the region of stable integration in the complex plane as K increases, and the requirement to store more time levels $u^{(n-q)}$ in order to reach $u^{(n+1)}$. For $K \leq 2$, (14) is A-stable [17, §8.5.4]; the default value in *Semtex* is $K = 2$.

Substituting (14) into (13) produces a stiffly-stable time integration scheme for the incompressible Navier–Stokes equations [3,11, §8.2.3.5];

$$\mathbf{u}^* = - \sum_{q=1}^K \alpha_q \mathbf{u}^{(n+1-q)} - \Delta t \sum_{q=0}^{K-1} \beta_q [\mathbf{N}(\mathbf{u}^{(n-q)}) - \mathbf{f}^{(n-q)}], \quad (15)$$

$$\nabla^2 P^{(n+1)} = (\Delta t)^{-1} \nabla \cdot \mathbf{u}^*, \quad (16)$$

$$\mathbf{u}^{**} = \mathbf{u}^* - \Delta t \nabla P^{(n+1)}, \quad (17)$$

$$\nabla^2 \mathbf{u}^{(n+1)} - \frac{\alpha_0}{\nu \Delta t} \mathbf{u}^{(n+1)} = - \frac{\mathbf{u}^{**}}{\nu \Delta t}, \quad (18)$$

where the weights α_q are those introduced in (14), and weights β_q are those for explicit polynomial-based extrapolation of values from time levels $(n-q)$ to time level $(n+1)$. *Semtex* uses equal-order approximations for velocity and pressure variables; it is a $\mathbb{P}_N - \mathbb{P}_N$ scheme. From the explicit-update step (15) we note the requirement for multi-level storage of $\mathbf{u}^{(n-q)}$ and $[\mathbf{N}(\mathbf{u}^{(n-q)}) - \mathbf{f}^{(n-q)}]$ if $K > 1$, along with an expectation of CFL-type instability if Δt is made too large. However, the viscous update (18) is implicit in time, so there is no problem with diffusion-related conditional instability of the time-splitting method.

A subtlety of this integration scheme as outlined in [11] is the use of the identity $\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u})$ when forming ‘high-order’ boundary conditions for the pressure-Poisson equation (16) from the Navier–Stokes equations (13). This leads to the following approximation for a computed-Neumann pressure boundary condition at time level $(n+1)$, on any boundary where the pressure is not otherwise available:

$$\begin{aligned} \partial_n P^{(n+1)} &\approx -\mathbf{n} \cdot \sum_{q=0}^{K-1} [\beta_q \mathbf{N}(\mathbf{u}^{(n-q)}) \\ &+ \nu \nabla \times \nabla \times \mathbf{u}^{(n-q)} + \partial_t \mathbf{u}^{(n-q)}]. \end{aligned} \quad (19)$$

With introduction of this boundary condition, the overall level of accuracy for the scheme is the same as for (14). The scheme is categorised as one of a class of rotational-form velocity-correction fractional-step projection methods by [12,13], who provide detailed discussion of stability and convergence properties. See also [3, §8.3.2].

Semtex also allows advection of a scalar variable c , in which case the Navier–Stokes equations are augmented by the advection–diffusion equation

$$\partial_t c + \mathbf{u} \cdot \nabla c = \alpha \nabla^2 c, \quad (20)$$

where α is the diffusion coefficient of species c . Evolution of c is handled by a straightforward extension to (15)–(18) with an explicit update step for advection of c included in (15) and an implicit treatment for diffusion of c included in (18).

As alluded to above, there are various ways of forming the advection terms besides the non-conservative forms $\mathbf{u} \cdot \nabla \mathbf{u}$ and $\mathbf{u} \cdot \nabla c$; in a continuum setting these are exactly equivalent to the conservative forms $\nabla \cdot (\mathbf{u}\mathbf{u})$ and $\nabla \cdot (\mathbf{u}c)$. However, the two formulations are not exactly equivalent in the discrete setting. In fully spectral DNS codes, it is well known that the ‘skew-symmetric’ form $[\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u})]/2$ has favourable properties in implicitly reducing aliasing errors when such products are formed [18] and that the cheaper ‘alternating’ form [19] in which the two alternatives are used on successive time steps performs almost as well. *Semtex* provides the alternating skew symmetric construction as the default, with full skew symmetric and non-conservative forms as options. No explicit dealiasing is employed,

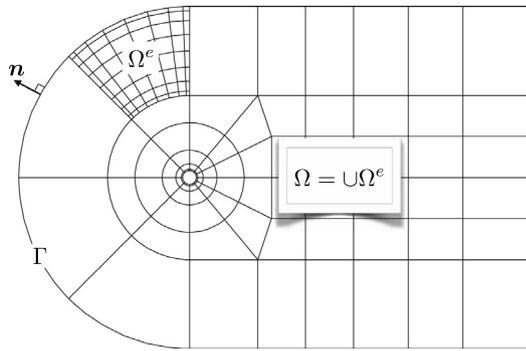


Fig. 5. A two-dimensional mesh that tessellates Ω as the union of conforming isoparametrically mapped elements Ω^e .

and time-integration in *Semtex* is quite robust to the effects of spatial under-resolution – see [20] for a discussion of the sources of aliasing errors and explicit dealiasing in the context of spectral element methods.

We note that the set of operations (15)–(18) amounts to updates (15) and (17) that may be computed locally on an element-by-element basis using the methods introduced in Section 2.1, together with solution of scalar elliptic equations: (16) is a Poisson equation for pressure, while (18) can be taken as a sequence of Helmholtz equations for the components of \mathbf{u} (and optionally, c). These elliptic equations are global in nature; in *Semtex* they are solved via continuous-Galerkin method of weighted residuals (MWR), a standard finite-element technique.

2.3. Global operations for solution of elliptic PDEs

With the details of element basis functions, their derivatives and integrals now defined, and motivated by the temporal splitting (15)–(18) of the Navier–Stokes equations, we turn to dealing with assemblies of elements, basis functions and how these are used to solve global elliptic PDEs on the domain $\Omega = \cup \Omega^e$, with boundary Γ which has unit outward normal \mathbf{n} , see Fig. 5.

Consider solving a Helmholtz problem $\nabla^2 u - \lambda^2 u = f$ in Ω , where λ is a real constant and f may be a function of space; successively setting λ and f to zero, this equation also becomes a model for dealing with Poisson and Laplace equations. The solution and its expansion in terms of basis functions is taken to be continuous across element boundaries. For a continuous (Bubnov-)Galerkin MWR solution, the elliptic PDE is multiplied by a weight function w drawn from the same basis set as the approximation for u , integrated over the domain Ω , and then treated using integration by parts to produce the ‘weak form’ of the PDE [5],

$$\int_{\Omega} (\nabla u \cdot \nabla w + \lambda^2 u w) \, d\Omega = - \int_{\Omega} f w \, d\Omega + \int_{\Gamma_N} h w \, d\Gamma \quad (21)$$

where $h \equiv \partial_n u \equiv \mathbf{n} \cdot \nabla u$ on parts Γ_N of Γ on which Neumann boundary conditions are applied. The weight function is taken as $w = 0$ on parts of the boundary Γ_D where Dirichlet boundary conditions are to be applied (where the values of $u_g \equiv g$ are supplied), thus ‘lifting’ the associated nodal values and shape functions out of the solution. Two key features of (21) are that the differentiability requirement on the solution (and shape functions) is reduced from two to one (formally they must reside in H^1), and that the first integral is symmetric in u and w , a

feature that carries over to the corresponding discrete statement (Helmholtz matrix).

To create a discrete form of (21) we commence by approximating $u(\mathbf{x})$ by a sum over a set of global basis functions $\mathcal{N}_j(\mathbf{x})$ multiplied by discrete coefficients u_j where j are global function and solution variable indices and

$$\begin{aligned} u(\mathbf{x}) &\approx \sum_{j=1}^Q u_j \mathcal{N}_j(\mathbf{x}) \\ &= \sum_{j=1}^P u_j \mathcal{N}_j(\mathbf{x}) + \sum_{j=P+1}^Q u_j \mathcal{N}_j(\mathbf{x}) \end{aligned} \quad (22)$$

where the partition $P < j \leq Q$ specifies the set of lifted functions that satisfy (known) Dirichlet boundary conditions, and we need to solve for the unknowns u_j , $1 \leq j \leq P$. The same set of functions \mathcal{N} is used to expand w and without loss of generality we can specify $w_j = 1$, $1 \leq j \leq P$ with $w_j = 0$ for $P < j \leq Q$. Inserting (22) into (21) gives the set of equations

$$\begin{aligned} \sum_{j=1}^Q u_j \int_{\Omega} [\nabla \mathcal{N}_j \cdot \nabla \mathcal{N}_i + \lambda^2 \mathcal{N}_j \mathcal{N}_i] \, d\Omega &\equiv \sum_{j=1}^Q H_{ij} u_j \\ &= - \int_{\Omega} f_j \mathcal{N}_j \mathcal{N}_i \, d\Omega + \int_{\Gamma_N} h_j \mathcal{N}_j \mathcal{N}_i \, d\Gamma, \end{aligned} \quad (23)$$

where H_{ij} is a global-system Helmholtz matrix. Terms of the form $\int \nabla \mathcal{N}_j \cdot \nabla \mathcal{N}_i \, d\Omega$ are typically referred to as ‘stiffness matrix’ contributions, while those of the form $\int \mathcal{N}_j \mathcal{N}_i \, d\Omega$ are typically called ‘mass matrix’ contributions, both names reflecting their origins in the solid mechanics community.

A graphical representation of (23), and how it may be rearranged for solution, is presented in Fig. 6, see also [3, §4.2.4.2]. We reiterate that Q is the number of global variables for a scalar field; the number of unknowns, P , is potentially lower, since there may be a number of locations at which Dirichlet data u_g are supplied on the boundary partition Γ_D .

To this point the set of global basis functions \mathcal{N}_i has been left unspecified. Now, finite element concepts are re-introduced, using a set of edge-conforming elements to tessellate the domain, $\Omega = \cup \Omega^e$, as indicated in Fig. 5, and isoparametric mappings of the elemental basis functions (2). The ‘global’ basis functions possess only comparatively local support, either, for functions that have only interior support within a single element, entirely confined within an element, or, for those with partial exterior support, spanning a small number of elements which mate along edges or at vertices. This idea is illustrated for a pair of one-dimensional elements in Fig. 7, and, for a pair of two-dimensional elements, in Fig. 4.14 of [3].

Another key finite element idea is that integrals across the whole domain Ω may be written as the sum of local integrals within elements, i.e.

$$\int_{\Omega} () \, d\Omega = \sum_{e=1}^{N_{el}} \int_{\Omega^e} () \, d\Omega^e, \quad (24)$$

where this summation is often called an ‘assembly’ operation, or, again reflecting origins in solid mechanics, a ‘direct stiffness’ summation. To carry it out in discrete form requires that, at least for global solution nodes u_j that are shared between elements, we have a single global numbering scheme. These concepts are discussed at some length in [3, §4.2] and more briefly in [2, §4.5.1]. We note in passing that *Semtex* uses Reverse Cuthill–McKee ordering [21] of element-boundary global node numbers in order to minimise the bandwidth of the assembled global Helmholtz matrix H_{uu} indicated in Fig. 6.

$$\begin{array}{c}
 \begin{array}{c}
 j \rightarrow \begin{array}{c} P \\ Q \end{array} \\
 \begin{array}{c} H_{uu} \\ H_{ug} \end{array} \begin{array}{c} \left\{ u \right\} \\ \left\{ u_g \right\} \end{array} = - \begin{array}{c} \left[M_{uu} \right] \begin{array}{c} \left\{ f \right\} \\ \left\{ h \right\} \end{array} + \begin{array}{c} \left\{ h \right\} \\ \left\{ \text{Neumann BCs} \right\} \end{array} \\
 \text{Dirichlet BCs} \rightarrow \left\{ u_g \right\} \\
 \text{Forcing}
 \end{array} \\
 \\
 \begin{array}{c}
 \begin{array}{c} \left[H_{uu} \right] \begin{array}{c} \left\{ u \right\} \\ \left\{ u_g \right\} \end{array} = - \begin{array}{c} \left[M_{uu} \right] \begin{array}{c} \left\{ f \right\} \\ \left\{ h \right\} \end{array} + \begin{array}{c} \left\{ h \right\} \\ \left\{ \text{Neumann BCs} \right\} \end{array} - \begin{array}{c} \left[H_{ug} \right] \left\{ u_g \right\} \end{array} \\
 \text{Cholesky}
 \end{array}
 \end{array}
 \end{array}$$

Fig. 6. The global Helmholtz equation (23) and how it may be rearranged for solution. The lower panel indicates that the global Helmholtz matrix H_{uu} is (banded, possibly block-diagonal) Cholesky, and that, for nodal spectral elements and equal-order GLL integration, the global mass matrix M_{uu} is, like equivalent elemental mass matrices, diagonal. Here, the index i traverses unknown variables and associated shape functions, while j traverses weight functions.

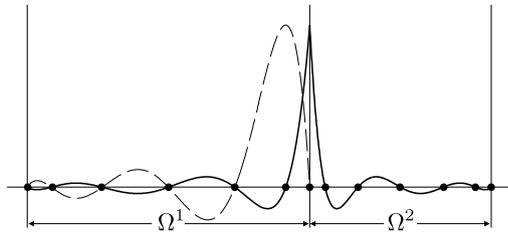


Fig. 7. Illustrating, on a pair of one-dimensional elements, that shape functions \mathcal{N} may either have support confined within a single element (dashed line), or extending across two mating elements (solid line).

Elemental contributions to the global Helmholtz matrix may be written

$$\begin{aligned}
 H_{ab}^e &= \int_{\Omega^e} [\nabla \mathcal{N}_b \cdot \nabla \mathcal{N}_a + \lambda^2 \mathcal{N}_b \mathcal{N}_a] d\Omega^e \\
 &\equiv \int_y \int_x [\nabla_{xy} \mathcal{N}_b \cdot \nabla_{xy} \mathcal{N}_a + \lambda^2 \mathcal{N}_b \mathcal{N}_a] dx^e dy^e
 \end{aligned}$$

but we wish to compute these on the standard domain $\bar{\Omega}^e$. This requires that we re-introduce the Jacobian of the isoparametric mapping (9) along with the original tensor-product elemental shape functions (2), producing

$$H_{ab}^e = \int_{-1}^1 \int_{-1}^1 [J^{-1} \nabla_{rs} \psi_b \cdot J^{-1} \nabla_{rs} \psi_a] |J| + \lambda^2 \psi_b \psi_a |J| dr ds, \quad (25)$$

where gradients ∇_{rs} are approximated using the derivative operator matrices D and D^T defined in (8) and integrals are approximated using GLL quadrature. For details, consult [3, §4.1], [22] or [23].

We note that in taking the weak form of the original PDE on the domain Ω , boundary integral terms (those on Γ_N) arose. The same considerations apply in taking the global assembly, however, the contributions across element edges are assumed to cancel out within the domain owing to the fact that the element-edge unit outward normals \mathbf{n} exactly oppose one another along mating edges, leaving only the global domain boundary terms that appear in both (21) and (23). Again, this is a standard finite element idea.

So far we have not discussed solution of the global Helmholtz equation (23). In *Semtex* there are two possibilities. The first

is direct solution based on Cholesky decomposition, but using element-level static condensation [a.k.a. Schur complement method or substructuring, see3, §4.2.3] as well as bandwidth minimisation in order to reduce the size of the assembled matrix H_{uu} which is used to first compute element-boundary values, following which back-substitution is used to find element-internal values. The second option uses iterative Jacobi- (i.e. diagonal-)preconditioned conjugate gradient techniques in which tensor-product forms are used to save operation counts [3, §4.1.6]. For either option, the contribution of $H_{ug}u_g$ to the right-hand-side vector is made using element-level operations and summation. The direct method can often provide faster solution, but at the cost of increased memory requirements compared to the iterative method. Since the pressure Poisson equation is poorly conditioned, iterative solution for the pressure step (16) is almost always inadvisable unless an effective pre-conditioner is available. On the other hand, the viscous Helmholtz equations become, owing to the term $\alpha_0/(v\Delta t)$ in (18), increasingly diagonally dominant and hence well-conditioned as viscosity falls, so that for high Reynolds number solutions, the iterative approach to the viscous solve may be both faster and cheaper in terms of memory requirements than direct solution.

2.4. Spectral element–Fourier discretisation

The treatment until this point has concentrated on two-dimensional domains and discretisations. As indicated in Section 1 and Fig. 1, *Semtex* can simulate flows in three-dimensional domains that are obtained by extrusion of two-dimensional domains along an orthogonal direction in which the solution is taken as periodic. Such flows are sometimes referred to as $2\frac{1}{2}$ -dimensional. In these cases, Fourier expansions are used in the out-of-plane direction, and the solution may, if required, be carried out in parallel across the resulting two-dimensional complex Fourier modes. If the length of the domain in the z direction is L_z we have, with $\beta = 2\pi/L_z$ and Fourier mode index k

$$\begin{aligned}
 \hat{\mathbf{u}}_k(x, y) &= \frac{1}{2\pi} \int_0^{L_z} \mathbf{u}(x, y, z) \exp(-ik\beta z) dz, \\
 \mathbf{u}(x, y, z) &= \sum_{k=-\infty}^{\infty} \hat{\mathbf{u}}_k(x, y) \exp(ik\beta z), \quad (26)
 \end{aligned}$$

where in practice only a finite number of Fourier modes $\hat{\mathbf{u}}_k$ are used, and, since the data \mathbf{u} are real, the Fourier data $\hat{\mathbf{u}}_k$ are

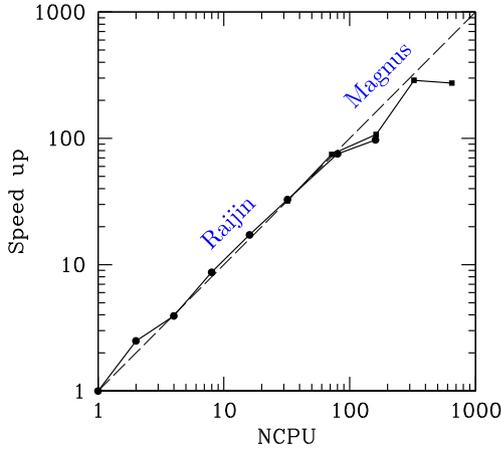


Fig. 8. Hard scaling (speed-up for problem of a fixed size) relations for Fourier-parallel DNS of a turbulent pipe flow obtained at two different supercomputer facilities. Speed up is approximately linear with number of processes until communications overheads become dominant.

conjugate-symmetric about $k = 0$, and only modes with $k \geq 0$ are retained. Hence, the number of complex modes is half the number of planes of real data, and a real-complex FFT based on [24] is used to effect the transformation to and from Fourier space. With the relationship above, the following hold for derivative operations in a Cartesian coordinate system

$$\frac{\partial \hat{u}_k}{\partial z} \equiv i\beta k \hat{u}_k, \quad \frac{\partial^2 \hat{u}_k}{\partial z^2} \equiv -\beta^2 k^2 \hat{u}_k, \quad (27)$$

where the operations can be done on a mode-by-mode basis owing to the linearity of the Fourier transform and of differentiation. From the second of these, and again for a Cartesian coordinate system,

$$\begin{aligned} \nabla^2 \hat{u}_k &\equiv \frac{\partial^2 \hat{u}_k}{\partial x^2} + \frac{\partial^2 \hat{u}_k}{\partial y^2} - k^2 \beta^2 \hat{u}_k \\ &\equiv \nabla_{xy}^2 \hat{u}_k - k^2 \beta^2 \hat{u}_k. \end{aligned} \quad (28)$$

For three-dimensional simulations, the Navier–Stokes equations are dealt with in Fourier-transformed space for most of the steps (15)–(18). The exception is during the formation of the nonlinear product terms $\mathbf{N}(\mathbf{u})$, which, as noted in Section 2.2, are computed without dealiasing; to reduce operation counts the associated variables are inverse-transformed to physical space, derivatives and products are formed, with the results finally forward transformed back to Fourier space. In parallel operations, such ‘pseudospectral’ operations require intra-process rearrangement of data, followed by block transposes which are carried out using an all-to-all exchange in MPI. Some of the details are outlined in [25]. The nature of speed up which can be obtained in parallel operations is indicated in Fig. 8; while specific outcomes are problem- and machine-dependent, the speed up which *Semtex* obtains is typically approximately linear with number of processes until transpose blocks become small enough that communications overheads dominate proceedings. In general, this saturation threshold increases with the number of elements and planes of data in the problem being computed.

Depending on the set of pressure boundary conditions selected, it is possible that the global pressure Poisson matrix for mode $k = 0$ can be singular. In this case, the pressure is set to zero for this mode at the highest globally numbered node in the domain, thus constraining the pressure field and removing the singularity.

2.5. Formulation for cylindrical coordinates

Problems set in cylindrical coordinate systems naturally have periodicity in the azimuthal direction and so are well suited to $2\frac{1}{2}$ -dimensional computations with Fourier expansions in the azimuth, where, for an azimuthal periodic angle of 2π , $\beta = 1$. Nonetheless, such coordinate systems possess a geometric singularity at the axis, which requires care when undertaking numerical approximations. The reader is referred to [26,27] for the component form of the incompressible Navier–Stokes equations in cylindrical coordinates.

In [10] the approach adopted in *Semtex* to dealing with the issue of geometric singularity is described in detail. Essentially, it involves a relatively straightforward extension to the treatment outlined above, which relies for success on the observation that whenever possibly singular terms arise on the axis, they are unproblematic owing either to mode index or boundary conditions applied to the Fourier mode in question. The key changes to the exposition above are (a) when dealing with the viscous substep (18) the complex radial and azimuthal velocity variables are coupled, but in a way that decouples the cylindrical-coordinate form of the Helmholtz equations [28], (b) multiplication of the Navier–Stokes equations by radius, which leads to symmetry of the weak form of the elliptic equations and (c) an appropriate choice of Fourier-mode-dependent axial boundary conditions for the various velocity components and pressure. As explained in [10] these three techniques, when used in combination with a Galerkin MWR treatment and the standard spectral element–Fourier basis, are sufficient to produce exponential convergence of the method as applied to the three-dimensional Navier–Stokes equations.

As outlined in [10, §3.4], computation of forcing terms associated with the cylindrical equivalent of the pressure Poisson equation (16) required some further consideration. In cylindrical coordinates, with the radius-premultiplied Navier–Stokes equations, and with Fourier-transformed variables, the forcing terms for Fourier mode k in (16) become

$$(\Delta t)^{-1}(\partial_z r \hat{u}_k^* + r \partial_r \hat{v}_k^* + \hat{v}_k^* + ik \hat{w}_k^*), \quad (29)$$

where, here, z and r are axial and radial coordinates in the meridional semi-plane (equivalent to x and y coordinates in the Cartesian treatment) and u , v and w are respectively axial, radial and azimuthal components of velocity. It may be seen that the two final terms here do not include the radius, so these might become non-zero at the cylindrical axis. In fact, \hat{v}_0^* , because it may include contributions from nonlinear product terms resulting from cross-axial flows allowed in Fourier mode $k = 1$, can remain finite at the axis. Further, the axial boundary condition applied for the axisymmetric pressure, $\partial \hat{p}_0 / \partial n = 0$, allows this forcing term to have an effect, so it needs to be retained in the MWR computation. The outcome is that the forcing (29) causes no problem provided that this feature of the formulation is recognised and, in creating the weighted-residual form of (29), radius is not pre-multiplied into the quadrature weights. (If radius is pre-multiplied in, a *division* by radius will then be required in forming the MWR equivalent of (29), leading to problems with terms like \hat{v}_0^* that might legitimately be non-zero at the axis.) Elsewhere in the code, however, it is quite appropriate and non-problematic to use radius-premultiplied quadrature weights in order to save operations. The *Semtex* implementation accounts for these issues when computing solutions to the cylindrical-coordinate forms of (15)–(18).

In Section 4.1 we will show that the resulting method retains spectral accuracy in all coordinate directions for three-dimensional Navier–Stokes solutions, indicating that the problem of geometric singularity at the axis has been overcome. We note

that it is very straightforward for the analyst to select solution in cylindrical coordinates since the option is fully incorporated into the solvers; in the most basic cases, all that is needed is to set the flag `CYLINDRICAL=1` in the `<TOKENS>` section of the associated session file (see Section 3.1). If the cylindrical axis lies along a domain boundary, axis boundary conditions are indicated in the session file and the appropriate type (homogeneous Dirichlet or Neumann) is selected automatically depending on the variable and Fourier mode, see e.g. Section 4.3.

3. Operational features

The *Semtex* distribution is supplied with instructions for compilation of executable files; the user has the choice of in-source compilation via `make`, or out-of-source compilation via `cmake`. These instructions are provided in a `README` file in the top-level *Semtex* directory. We recommend taking the `cmake` option in the first instance. The key executables are `dns`, for DNS of the incompressible Navier–Stokes equations, and `elliptic`, a stand-alone solver for elliptic PDEs. If an MPI distribution is detected, Fourier-parallel versions of these two solvers, `dns_mp` and `elliptic_mp`, will also be compiled. In addition, the *Semtex* distribution comes with a set of (serial-only) utilities for pre- and post-processing of simulation results. A user guide, which gives in-depth explanation of session files and other supporting files types, utilities, and solver options, is supplied in the `doc` subdirectory.

Flow simulations in *Semtex* may be either two-dimensional, with number of z -planes, $N_Z = 1$, or three-dimensional ($N_Z \geq 4$). For two-dimensional simulations, either two or three velocity components may be chosen (e.g. an axisymmetric swirling flow simulation would employ a cylindrical coordinate system, a single plane of data described in two spatial coordinates, and three velocity components). For three-dimensional simulations, three velocity components are required.

3.1. Solver switches and user-defined variables

The values of various pre-defined *Semtex* solution switches and user-defined constants may be set at run time in the `<TOKENS>` section of a session file. In general, each line of this section is of the form `token = expression` where the expression is a string which is parsed to generate a numeric value that is stored internally in a lookup table. Each token thus evaluated is used to either set specific solver flags (if the token string has a predefined significance within *Semtex*) or may be re-used within the session file in the setting of boundary conditions, forcing terms, or initial conditions. The reader is referred to the user guide for further details.

3.2. Forcing terms

Semtex allows for the inclusion of various forcing terms $\mathbf{f}(\mathbf{x}, t)$ in the Navier–Stokes equations (13) including:

- Constant, \mathbf{f}
- Spatio-temporal, $+a(\mathbf{x})\alpha(t)$
- Sponge region, $-m_1(\mathbf{x})(\mathbf{u} - \mathbf{u}_0)$
- Drag, $-m_2(\mathbf{x})(\mathbf{u}/|\mathbf{u}|)|\mathbf{u}(\mathbf{x}, t)|^2$
- Selective frequency damping [29], $-\chi(\mathbf{u} - \bar{\mathbf{u}})$
- Steady and unsteady Coriolis, $-2\boldsymbol{\Omega} \times \mathbf{u} - (d\boldsymbol{\Omega}/dt) \times \mathbf{x} - \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x})$
- Boussinesq buoyancy $-\beta_T(c - c_{ref})\mathbf{g}$

Spatio-temporal, sponge region and drag forces may be specified via analytic functions using standard mathematical syntax within the `<FORCE>` section of a session file. These are evaluated by the yacc-based function parser included in *Semtex*.

```
<FORCE>
  SPATIOTEMP_ALPHA_X = cos(x)*step(t,10)
  SPATIOTEMP_ALPHA_Y = -sin(z)*step(t,10)
  SPATIOTEMP_ALPHA_Z = -cos(y)*step(t,10)
</FORCE>
```

For a complete description of the various forcing terms available in *Semtex* we refer the reader to §4.12 of the user guide included in the distribution.

3.3. Boundary and initial conditions

In addition to the high-order pressure boundary condition (19) and axial boundary conditions required for cylindrical geometries [10], boundary conditions must also be applied to the velocity and temperature fields. In the x and y directions these are typically Dirichlet/Neumann or inflow/outflow conditions. For outflow conditions, one common choice is an approximation to a stress-free boundary condition obtained by setting $\partial_n u = \partial_n v = \partial_n w = p = 0$, while another, more robust, possibility is formulated as in [30]. These boundary conditions may be tagged within the `<BCS>` section of a session file as D (Dirichlet), N (Neumann), H (high-order pressure condition), O (outflow [30]), and A (axial, see e.g. [10] and Section 4.3). For example:

```
<BCS NUMBER=1>
1 w 4
  <D> u = sin(TWOPI*x) </D>
  <D> v = 1.0 </D>
  <N> c = 0.0 </N>
  <H> p </H>
</BCS>
```

assigns, to the boundary tagged as `w`, Dirichlet boundary conditions for the u and v velocity components as 0.0 and 1.0 respectively, a homogeneous Neumann condition to the temperature c , and specifies the use of the high-order boundary condition for pressure p . Periodicity does not formally constitute a boundary condition and hence is not dealt with in the `<BCS>` section; in the (x, y) plane, periodicity is, instead, specified in the `<SURFACES>` section of a session file. Periodicity occurs naturally in the out-of-plane (z) direction owing to the adoption of Fourier expansions, and requires no explicit specification by the analyst.

As for the forcing terms, initial conditions may be specified via analytic functions. For example a two-dimensional Taylor vortex flow field may be initialised as

```
<USER>
  u = -cos(PI*x)*sin(PI*y)
  v = sin(PI*x)*cos(PI*y)
</USER>
```

3.4. Utilities

In addition to an incompressible Navier–Stokes solver and a stand-alone elliptic solver, the *Semtex* distribution also includes a suite of utilities for pre/post-processing and analysis. These include

- `addfield` Compute additional fields such as kinetic energy or enstrophy from the prognostic fields supplied in *Semtex* .fld output files.
- `integrate` Compute the global integrals of scalar variables in a given .fld file via GLL quadrature.
- `sem2tec` Generate a binary *Tecplot*-format .plt file from a given output file for visualisation, which can also be read in by *ParaView* and *VisIt*.

- meshpr Output the mesh geometry.
- enumerate Generate a global node numbering (a .num file).
- compare Generate a .fld file from user-defined analytical functions to be supplied as initial conditions, or optionally subtract the analytical functions from data in a given .fld file.
- convert Convert .fld file formats, extract a specific numbered dump from a catenated .fld file.
- probe Probe a .fld file at a set of spatial locations.
- calc Command-line-driven calculator based on the Semtex function parser.

For the complete set of *Semtex* utilities refer to §2.5 and Ch. 6 of the user guide.

4. Applications

All *Semtex* models are described to the solver in a session file, which details the relevant boundary and initial conditions, mesh geometry and model parameters for a given configuration. The name of this file is passed to the *Semtex* executable as an input argument at run time. Prior to running a given model in *Semtex* the mesh global numbering file session.num must be generated via the enumerate utility. Optionally, an initial condition (restart) file session.rst may also be generated e.g. via the compare utility, or as a result of a previous run: if this file is not present, all variables are initialised to zero values.

4.1. Kovaszny flow within an offset cylindrical domain

Kovaszny [31] provided a non-trivial two-dimensional, two-component steady solution of the Navier–Stokes equations, as follows:

$$\begin{aligned} u &= 1 - \exp(\lambda x) \cos(2\pi y), \\ v &= (2\pi)^{-1} \lambda \exp(\lambda x) \sin(2\pi y), \\ p &= (1 - \exp \lambda x)/2, \end{aligned}$$

where $\lambda = Re/2 - (Re^2/4 + 4\pi^2)^{1/2}$, $Re \equiv 1/\nu$. When considered in cylindrical coordinates, however, this can be re-cast as a three-dimensional, three-component solution. As outlined in [10] and illustrated in Fig. 9, if the axis of the cylindrical domain is offset from the x -axis as used by Kovaszny and in addition the domain is rotated in azimuth by angle θ , a non-rational multiple of π , all components of the cylindrical-coordinate form of the Navier–Stokes equations are energised, and the problem becomes an excellent test case for cylindrical-coordinate Navier–Stokes solvers.

The two test case meshes examined in [10] are provided in session files cylkov1 and cylkov2 in the *Semtex* distribution. As shown in Fig. 10, errors of the computed steady-state solution reduce exponentially fast in the asymptotic limits of increased spectral element and Fourier resolution until double-precision machine-noise limits are reached, thus demonstrating the anticipated spectral spatial convergence properties.

4.2. Transition and decay of a Taylor–Green vortex

The temporal evolution of kinetic energy dissipation ϵ for a freely decaying Taylor–Green vortex initial condition in a periodic box with side lengths $L_x = L_y = L_z = 2\pi$ has been widely used to validate methods to solve the incompressible Navier–Stokes equations [see e.g.32,33]. With initial condition

$$u = + \sin(x) \cos(y) \cos(z), \tag{30}$$

$$v = - \cos(x) \sin(y) \cos(z), \tag{31}$$

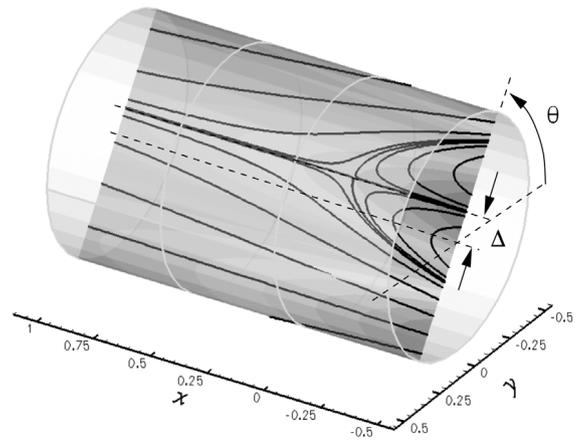


Fig. 9. The two-dimensional analytical solution of the Navier–Stokes equations provided by Kovaszny [31], considered in a cylindrical geometry. Source: From [10].

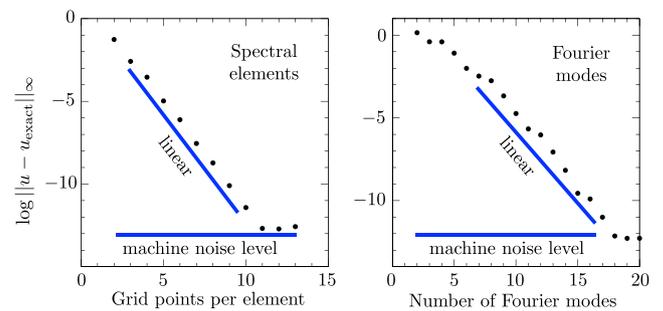


Fig. 10. Demonstrations of spectral convergence for the cylindrical-coordinate Kovaszny test case illustrated in Fig. 9. Source: From [10].

$$w = p = 0. \tag{32}$$

and with $\nu = 1/1600$, the equivalent Reynolds number $Re = 1600$ [32]. The kinetic energy dissipation is computed as $\epsilon = -dE/dt$ where

$$E = \frac{1}{2V} \int_V \mathbf{u} \cdot \mathbf{u} \, dV \tag{33}$$

for computational volume $V = L_x L_y L_z$. A *Semtex* session file for simulation for this problem is provided as TG1600; a grid of 32×32 spectral elements is used in the (x, y) plane, with tensor product shape functions of order $N \times N = 7 \times 7$ in each element, and 256 planes of data (i.e. 128 Fourier modes) are used in the z direction. The spatial resolution is approximately 256^3 . The simulation is carried out with 3rd-order time integration ($J = 3$, token N_TIME=3) and time step $\Delta t = 0.0025$ (token D_T \$= \$ 0.0025).

The initial condition is specified in the <USER> section of the session file

```
<USER>
  u = sin(x)*cos(y)*cos(z)
  v = -cos(x)*sin(y)*cos(z)
  w = 0
  p = 0
</USER>
```

and an initial condition file (TG1600.rst) was generated using the compare utility. Following completion of the simulation in which output dumps were concatenated into a single file at every

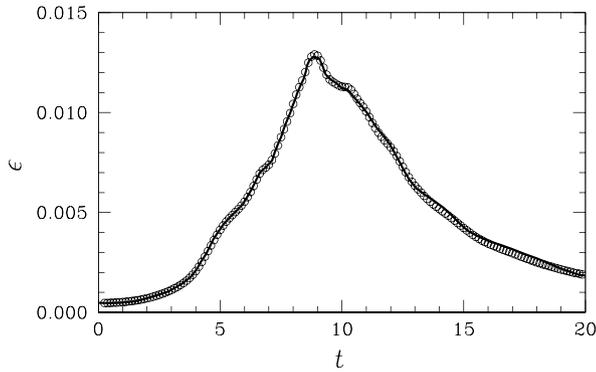


Fig. 11. Temporal evolution of kinetic energy dissipation ϵ for a Taylor–Green vortex initial condition. Solid line: 512^3 reference data [33], open symbols: *Semtex* results.

50 time steps, the kinetic energy was computed in post processing using the `addfield` utility and integrated over the domain volume using the `integrate` utility. Subsequently the dissipation was estimated using central differencing. The outcome is shown in Fig. 11, with results from a 512^3 Fourier-spectral reference simulation [33] shown for comparison.

4.3. Oscillatory flow past a sphere

Heat or mass transfer from spherical particles in oscillatory flow is found in applications such as combustion and spray drying. This problem was investigated in [34] where an isolated sphere with uniform surface concentration of scalar c was considered in a flow with rectilinear far-field oscillation. In the following, we briefly describe the workflow for solving this problem using *Semtex*. The configuration for this case is provided in the session file `sphere05`.

The computational domain represents a sphere of diameter $D = 1$ placed at the centre of a cylindrical domain with axial, radial coordinates (x, r) as shown in Fig. 12(a). The associated velocity variables are (u, v) . The three segments of the domain boundary which do not intersect the sphere represent the far-field boundary while the fourth is the cylindrical coordinate axis. The domain extents are $x_{\max}/D = \pm 50$ and $r_{\max}/D = 50$, which are large in order to reduce interference of the far-field boundary on the flow dynamics near the sphere. Fig. 12 shows a sample mesh where spectral elements with $N = 8$ are used to discretise the domain. An extremely fine radial mesh near the sphere was used to capture the dynamics at low oscillation amplitudes and high Re .

The hydrodynamics of the flow is characterised via the amplitude ratio A/D with A being the amplitude of freestream fluid displacement and Reynolds number $Re = U_{\max}D/\nu$ where $U_{\max} = \omega A$ is the maximum freestream velocity and ω is the frequency of rectilinear oscillations. The governing equations (13) and (20) are solved in cylindrical coordinates (by setting token `CYLINDRICAL=1`) and the flow is assumed axisymmetric. A passive scalar c representing mass concentration or temperature is added to the system, and this is non-dimensionalised with respect to its maximum value, c_{\max} , located at the sphere wall.

No-slip boundary conditions are specified for the velocity at the sphere wall and, at the far-field boundary, $u = U_{\max} \cos \omega t$, $v = 0$. At the axis, the appropriate boundary conditions are [10] $\partial_n u = v = \partial_n c = \partial_n p = 0$ where n is the axis-normal direction (but note: these values are set internally by the solver). The viscosity (`KINVIS`) is set to U_{\max}/Re and the species diffusivity α is assumed to be equal to the kinematic viscosity (by setting token `PRANDTL=1`). The boundary conditions are specified in the `<BCS>` section of the session file as:

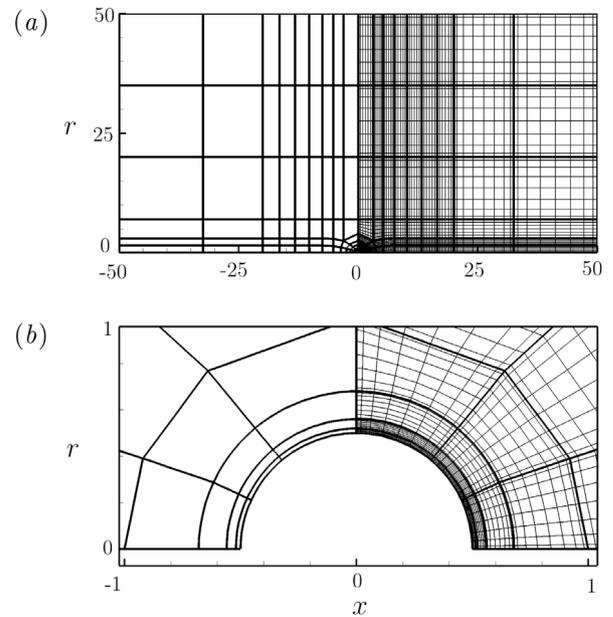


Fig. 12. (a), spectral element mesh (left) and GLL nodes (right) for oscillatory flow past a sphere. A close-up view of the mesh near the sphere is shown in (b).

```
<BCS NUMBER=3>
1 f 4
  <D> u = UMAX*cos(OMEGA*t) </D>
  <D> v = 0.0 </D>
  <D> c = 0.0 </D>
  <H> p </H>
2 a 4
  <A> u </A>
  <A> v </A>
  <A> c </A>
  <A> p </A>
3 w 4
  <D> u = 0.0 </D>
  <D> v = 0.0 </D>
  <D> c = 1.0 </D>
  <H> p </H>
</BCS>
```

where `f` is the BC tag used for the farfield boundary, `a` for the axis and `w` for the sphere wall. `UMAX` and `OMEGA` are specified in the `<TOKENS>` section. Simulations can be initialised with zero velocity and scalar field and the time-averages of the field variables are collected by setting the token `AVERAGE=1` and integrating over a long time. Fig. 13 shows the time-averaged scalar concentration at $x/D = 0$ for various Re where it can be seen that as Reynolds numbers decrease, the time-averaged concentrations asymptote, as expected, towards an analytical solution for Stokes flow.

4.4. Turbulent pipe flow

Turbulent pipe flow of Newtonian fluid is a canonical problem of fluid dynamics and has been studied for many decades owing to its relevance in a wide range of applications. Owing to the presence of two homogeneous directions, pipe flows may be simulated in *Semtex* using either Cartesian or cylindrical coordinate-based discretisations. In the following we briefly discuss the Cartesian-coordinate simulations of [35,36]; for cylindrical-coordinate simulations of turbulent pipe flows consult [37,38].

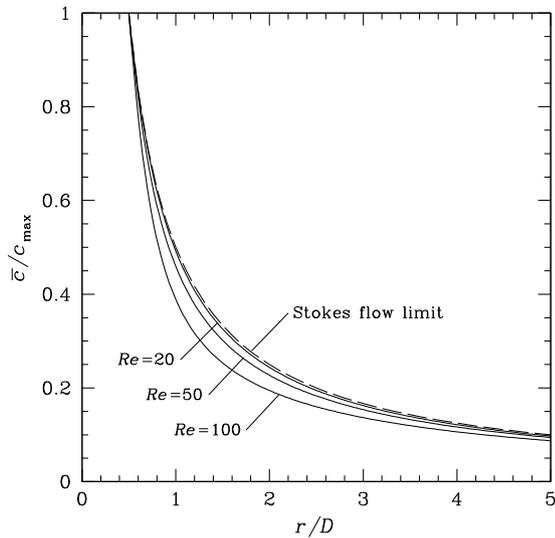


Fig. 13. Time-average concentration profiles for oscillatory flow past a sphere plotted at $x/D = 0$ for $A/D = 0.05$ and $Re = 20-100$ and $Pr = 1$. Source: Reproduced from [34].

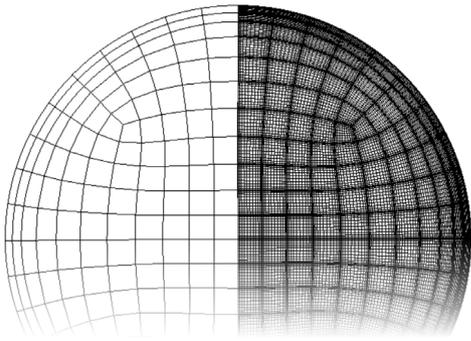


Fig. 14. A spectral element mesh used to simulate turbulent pipe flow [35,36]. Spectral element boundaries are shown on the left and grid nodes for 12th-order tensor-product element interpolation functions are shown on the right. Fourier expansions are used in the pipe-axial direction.

The configuration to solve the flow in Cartesian coordinates is provided in the session file `pipe323` in which the pipe cross-section is discretised using the spectral elements and Fourier expansions are used in the axial direction. Pipe length L_z is implied via the token `BETA`; $L_z = 2\pi/BETA$. Sufficient mesh resolution is required in the axial, azimuthal and the radial directions; guidelines for wall-resolving spectral element mesh designs are discussed in [39]. Fig. 14 shows a cross-section of a *Semtex* pipe mesh discretisation.

Flow is driven by a constant body force \mathbf{f} ; this allows the pressure to be periodic in the pipe-axial direction as required for Fourier expansions. For a given time-average friction Reynolds number $Re_\tau = u^*R/\nu$ (where $u^* = (\tau_w/\rho)^{1/2}$ and R is pipe radius) the required axial body force per unit mass is [38]

$$f_z = 2Re_\tau^2 \nu^2 / R^3.$$

This is set in the `<FORCE>` section via the expression `CONST_Z=FFZ`, with the value `FFZ` pre-calculated in the `<TOKENS>` section according to the above formula and with $Re_\tau = 323.89$.

Simulations can be initialised with a laminar velocity profile perturbed using the `noiz` utility to add white noise in the initial conditions, as is often required to help initialise turbulent flow simulations. During simulation, the energy in the Fourier

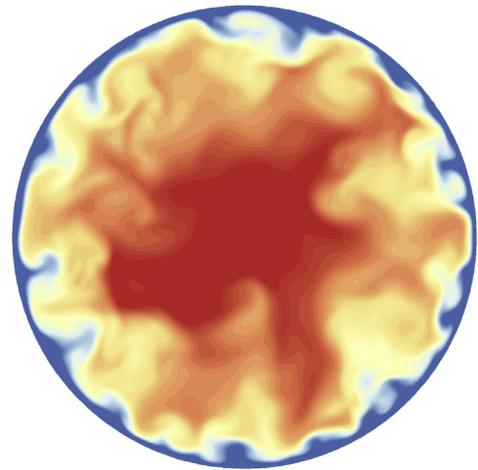


Fig. 15. Contours of instantaneous axial velocity on a cross section from DNS of turbulent pipe flow at $Re_\tau = 323$ [35,36].

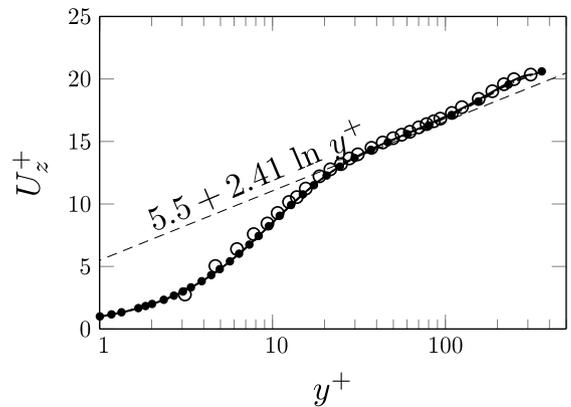


Fig. 16. Comparison of the mean axial velocity profiles from a *Semtex* simulation of turbulent pipe flow at $Re_\tau = 323$ (solid line, [36]) plotted in wall units and compared to experimental measurements of [40] at $Re_\tau = 314$ (open circles) and DNS data of [41] at $Re_\tau = 360$ (filled circles).

wavenumbers $k > 0$ grows for a turbulent flow, and once saturated to a statistically steady state, these values fluctuate with simulation time, indicating a non-axisymmetric unsteady flow. At this stage, the total viscous force per unit length also fluctuates somewhat about the mean value $\pi/4 \times f_z$. This can be confirmed from the integral of the viscous forces over the wall reported in the `pipe323.flx` file at every `IO_HIS` steps. *Semtex* writes the integral of the viscous and pressure forces in the `pipe323.flx` file only for the `wall` group defined in the `<GROUPS>` section (consult Chs 2 and 4 of the user guide).

Fig. 15 shows contours of instantaneous axial velocity extracted at a pipe cross-section after the flow has become statistically stationary; a range of eddy length scales may be seen. After saturation, the time-averages of flow field variables can be collected to file `pipe323.avg` by setting the token `AVERAGE` to 1, 2 or 3 as desired until the averages become time-invariant, which takes approximately twelve to fifteen mean-flow transit times of the domain. Subsequently the results are also averaged in the axial and azimuthal directions to further reduce statistical variation and then presented as functions of distance from the wall. Fig. 16 shows the mean axial velocity profile predicted by *Semtex* compared with experimental measurements and DNS results obtained using another code.

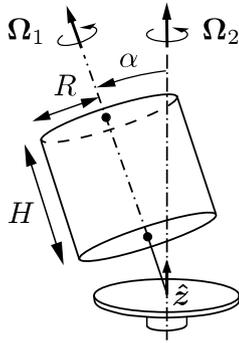


Fig. 17. Schematic of experimental setup for examination of flow in a precessing cylinder [42,43]. The cylinder spins steadily in a tilted gimbal mounted on a rotating turntable.

4.5. Instability within a precessing cylinder

As a final demonstration we present results for the triadic resonance of modes in a rotating, precessing cylinder [42,43]. The physical system under consideration is illustrated schematically in Fig. 17. The system is forced by the Coriolis term as given in Section 3.2 with $\Omega = \Omega_1 + \Omega_2$, where Ω_1 is the angular velocity of the cylinder and Ω_2 is the angular velocity of precession.

The cylinder rotation rate $\Omega_1 = |\Omega_1| \text{sgn}(\Omega_1 \cdot \hat{z})$ is taken as positive and the turntable rotation rate $\Omega_2 = |\Omega_2| \text{sgn}(\Omega_2 \cdot \hat{z})$ may have either sign. The system is described by four dimensionless groups which may be taken as the precessional nutation (tilt) angle α , the cylinder aspect ratio $\Gamma = H/R$, the nondimensional angular frequency ratio $\omega_F = \Omega_1/(\Omega_1 + \Omega_2 \cos \alpha)$ and the Reynolds number $Re = \Omega_1 R^2/\nu$. A session file associated with the computations of [43] for $\alpha = 3^\circ$ is supplied as `PreCy103`; as outlined in [43] the simulation is carried out in the gimbal (precessing) frame of reference in which Ω is steady in time and a steady basic state may be computed using selective frequency damping (see Section 3.2). In this frame of reference the cylinder rotates steadily at rate Ω_1 , and this rotation component is thus dealt with in the <BCS> section of `PreCy103`. The DNS was run in a cylindrical geometry with 16×12 elements of order $N \times N = 6 \times 6$, and 64 Fourier modes in the azimuthal direction.

The basic state, dominated by the $m = 1$ azimuthal Fourier mode, is associated with a three-dimensional overturning circulation in the cylinder. Following deactivation of selective frequency damping, weakly nonlinear interactions between this mode and higher harmonics leads to the parasitic growth of these higher modes as shown in Fig. 18, [43, Fig. 5], whereby they draw energy from the basic state via a triadic-resonance instability, which saturates at later times owing to viscous and nonlinear effects. In the gimbal frame of reference, this instability manifests as a Hopf bifurcation.

The Coriolis acceleration term for rotation component Ω_2 (which appears steady in the gimbal frame) and selective frequency damping forcing terms are specified within the session file as:

```
<FORCE>
CORIOLIS_UNSTEADY = 0

CORIOLIS_OMEGA_X = OMEGA2*cos(THETA_MAX)
CORIOLIS_OMEGA_Y = 0.0
CORIOLIS_OMEGA_Z = OMEGA2*sin(THETA_MAX)

SFD_DELTA = SFD_PERIOD/TWOPI
SFD_CHI = (1-heav(t-SFD_T1))*0.1
</FORCE>
```

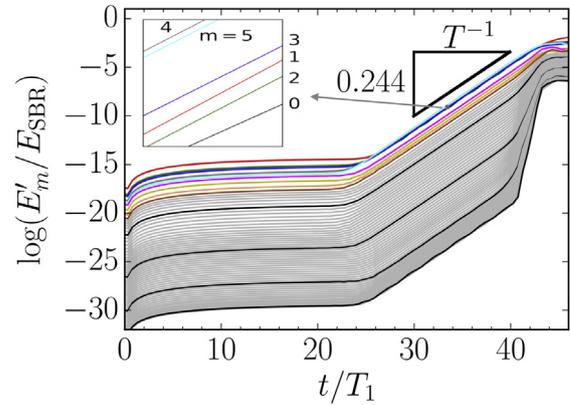


Fig. 18. Evolution of azimuthal Fourier modal energies with time for $\Gamma = 2.667$, $\omega_F = 0.735$, $\alpha = 3.0^\circ$ and $Re = 4778$ for flow in a precessing cylinder developing from a quasi-steady basic state. [43]. T_1 is the cylinder rotation period and E_{SBR} is the kinetic energy of solid-body rotation. Energies in the leading azimuthal modes during an exponential growth phase (inset) reflect a triadic resonance instability.

Source: From [43].

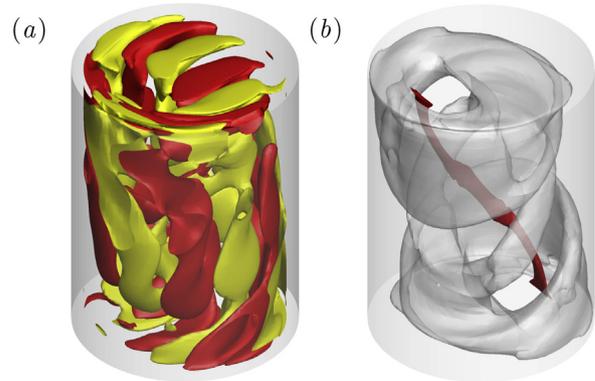


Fig. 19. (a) an instantaneous snapshot of a triadic resonance instability wave within a precessing cylinder, visualised in \pm perturbation axial vorticity and (b) an isosurface of (de-trended) time averaged kinetic energy perturbation (grey) and vortex core of the basic state (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Source: From [43].

Following attainment of an approximately steady basic state (prior to $t = 0$ in Fig. 18) selective frequency damping is deactivated via a Heaviside function so as to allow for the free evolution of higher modes through triadic resonance processes.

When considered in the cylinder frame of reference rather than the gimbal frame, DMD analysis [44] of simulation snapshots samples during the exponential growth phase of Fig. 18 shows a clear relationship between azimuthal Fourier indices of the DMD modes and their temporal frequencies that is consistent with a generalised triadic resonance mechanism for the instability [43]. The group velocity of the set of parasitic DMD modes is the same as that of the basic state as observed in the cylinder frame. Fig. 19 shows the nature of the instability as it grows within the cylinder. While the instability wave has a complicated spatio-temporal structure, the envelope of time-average kinetic energy (after de-trending for exponential growth) is steady and maintains a fixed relationship to the core of the basic state.

5. Discussion and conclusions

This article reviews the technical formulation and practical capabilities of the *Semtex* model for the direct numerical simulation

of the incompressible Navier–Stokes equations. The code is well suited to the simulation of flows within physical domains with one homogeneous dimension, which may be configured as either cylindrical or Cartesian geometries. For problems in such domains *Semtex* may exhibit significant performance improvements over traditional three dimensional codes due to the exact nature of differential operators in Fourier space, and the computational efficiency with which Fourier transforms may be applied.

Details are provided as to the modelling capabilities of *Semtex*, including the various forcing terms, boundary conditions and geometries that may be incorporated into model configurations. Examples are provided demonstrating these capabilities for a variety of physical problems, ranging from model validation tests to complex flows driven by temporally or spatially varying body forces. We also discuss the various additional utilities provided as part of the *Semtex* distribution for the generation of initial conditions and the analysis of model output.

Acknowledgements

The first author would like to thank Ron Henderson, whose *Prism* code formed the original basis for *Semtex*, for a great deal of help and encouragement during the beginnings of this project, and George Karniadakis for hosting the visit which enabled all the subsequent developments. Thanks are also due to Spencer Sherwin for many helpful and insightful discussions over the years, and to Chris Cantwell and Dave Moxey for practical help and suggestions.

References

- [1] A. Patera, *J. Comput. Phys.* 54 (1984) 468–488.
- [2] M.O. Deville, P.F. Fischer, E.H. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, 2002.
- [3] G.E. Karniadakis, S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, second ed., Oxford University Press, 2005.
- [4] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*, Springer, 2007.
- [5] T.J.R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, 1987, Dover edition, 2000.
- [6] C.H. Amon, A.T. Patera, *Phys. Fluids A* 1 (2) (1989) 2005–2009.
- [7] G.E. Karniadakis, *Comp. Meth. Appl. Mech. Eng.* 80 (1990) 367–380.
- [8] D. Chu, R.D. Henderson, G.E. Karniadakis, *Theoret. Comput. Fluid Dyn.* 3 (1992) 219–229.
- [9] C.H. Amon, *AIAA J.* 31 (1) (1993) 42–48.
- [10] H.M. Blackburn, S.J. Sherwin, *J. Comput. Phys.* 197 (2004) 759–778.
- [11] G.E. Karniadakis, M. Israeli, S.A. Orszag, *J. Comput. Phys.* 97 (1991) 414–443.
- [12] J.L. Guermond, J. Shen, *SIAM J. Numer. Anal.* 41 (2003) 112–134.
- [13] J.L. Guermond, P. Mineev, J. Shen, *Comput. Meth. Appl. Mech. Eng.* 195 (2006) 6011–6045.
- [14] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, 1964, Dover edition, 1965.
- [15] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Springer, 2006.
- [16] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey, 1971.
- [17] G. Dahlquist, Å. Björck, *Numerical Methods*, Prentice-Hall, 1974, Dover edition, 2003.
- [18] T.A. Zang, *Appl. Numer. Math.* 7 (1991) 27–40.
- [19] R.M. Kerr, *J. Fluid Mech.* 153 (1985) 31–58.
- [20] G. Mengaldo, D.D. Grazia, D. Moxey, P.E. Vincent, S.J. Sherwin, *J. Comput. Phys.* 299 (2015) 56–81.
- [21] A. George, J.W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.
- [22] E.M. Rønquist, Ph.D. thesis, Massachusetts Institute of Technology, 1988.
- [23] Y. Maday, A.T. Patera, in: A.K. Noor, J.T. Oden (Eds.), *State-of-the-Art Surveys on Computational Mechanics*, ASME, 1989, pp. 71–143.
- [24] C. Temperton, *SIAM J. Sci. Stat. Comput.* 13 (3) (1992) 676–686.
- [25] M. Rudman, H.M. Blackburn, *Appl. Math. Model.* 30 (2006) 1229–1248.
- [26] R.B. Bird, W.E. Stewart, E.N. Lightfoot, *Transport Phenomena*, Wiley, 1962, Second printing.
- [27] G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.
- [28] S.A. Orszag, A.T. Patera, *J. Fluid Mech.* 128 (1983) 347–385.
- [29] E. Åkervik, L. Brandt, D.S. Henningson, J. Hoepfner, O. Marxen, P. Schlatter, *Phys. Fluids* 18 (2006) 068102–1–4.
- [30] S. Dong, G.E. Karniadakis, C. Chrysosostomidis, *J. Comput. Phys.* 261 (2014) 83–105.
- [31] L.I.G. Kovaszny, *Proc. Camb. Phil. Soc.* 44 (1948) 58–62.
- [32] W.M. van Rees, A. Leonard, D.I. Pullin, P. Koumoutsakos, *J. Comput. Phys.* 230 (2011) 2794–2805.
- [33] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, *Int. J. Numer. Meth. Fluids* 72 (2013) 811–845.
- [34] H.M. Blackburn, *Phys. Fluids* 14 (2002) 3997–4011.
- [35] J. Singh, M. Rudman, H.M. Blackburn, *J. Fluid Mech.* 822 (2017) 848–879.
- [36] J. Singh, M. Rudman, H.M. Blackburn, *Phys. Rev. F* 3 (2018) 094607.
- [37] C. Chin, A.S.H. Ooi, I. Marusic, H.M. Blackburn, *Phys. Fluids* 22 (11) (2010) 115107.
- [38] S. Saha, J.C. Klewicki, A. Ooi, H.M. Blackburn, *J. Fluid Mech.* 779 (2015) 245–274.
- [39] H.M. Blackburn, S. Schmidt, *J. Comput. Phys.* 186 (2) (2003) 610–629.
- [40] J.M.J. den Toonder, M.A. Hulsen, G.D.C. Kuiken, F.T.M. Nieuwstadt, *J. Fluid Mech.* 337 (1997) 193–231.
- [41] G.K. El Khoury, P. Schlatter, A. Noorani, P.F. Fischer, G. Brethouwer, A.V. Johansson, *Flow Turb. Combust.* 91 (3) (2013) 475–495.
- [42] T. Albrecht, H.M. Blackburn, J.M. Lopez, R. Manasseh, P. Meunier, *J. Fluid Mech.* 778 (2015) R1–1–12.
- [43] T. Albrecht, H.M. Blackburn, J.M. Lopez, R. Manasseh, P. Meunier, *J. Fluid Mech.* 841 (2018) R3–1–13.
- [44] P.J. Schmid, *J. Fluid Mech.* 656 (2010) 5–28.