

The Openpipeflow Navier–Stokes solver

Ashley P. Willis

School of Mathematics and Statistics, University of Sheffield, South Yorkshire, S3 7RH, UK



ARTICLE INFO

Article history:

Received 5 February 2017

Received in revised form 14 May 2017

Accepted 16 May 2017

Keywords:

CFD

Dynamical systems

ABSTRACT

Pipelines are used in a huge range of industrial processes involving fluids, and the ability to accurately predict properties of the flow through a pipe is of fundamental engineering importance. Armed with parallel MPI, Arnoldi and Newton–Krylov solvers, the Openpipeflow code can be used in a range of settings, from large-scale simulation of highly turbulent flow, to the detailed analysis of nonlinear invariant solutions (equilibria and periodic orbits) and their influence on the dynamics of the flow.

© 2017 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version	1.12
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00015
Legal Code License	GNU General Public License
Code versioning system used	none
Software code languages, tools, and services used	Fortran90, MPI(optional)
Compilation requirements, operating environments & dependencies	NetCDF, LAPACK, FFTW3
If available Link to developer documentation/manual	http://www.openpipeflow.org=index.php?title=Manual
Support email for questions	ashleypwillis@gmail.com

1. Motivation and significance

The flow of fluid through a straight pipe of circular cross-section is a canonical setting for the study of stability, transition and properties of turbulent flow. At low flow rates, the flow everywhere is in the direction parallel to the axis of the pipe, a simple ‘laminar’ flow. At larger flow rates it typically undergoes a transition to a complex ‘turbulent’ flow, characterised by an abundance of swirling eddies. As early as 1883, Reynolds observed that the transition from laminar to turbulent flow is highly dependent on perturbations of finite amplitude to the initial flow [1].¹ Nevertheless, he also noticed that the appearance of turbulence is consistent with respect to the value of the non-dimensional combination DU/ν , at around 2000, where U is the mean axial speed, D the diameter of the pipe, and ν the kinematic viscosity. This combination is the now famous Reynolds Number, $Re = DU/\nu$, used in a huge range of systems involving fluids, where D and U are typical length and velocity scales for the system.

¹ E-mail address: ashleypwillis@gmail.com.

¹ Reynolds referred to what we now call ‘laminar’ and ‘turbulent’ flows by ‘direct’ and ‘sinuous’ flow, respectively.

It has been known for some time that the Navier–Stokes equations together with the no-slip boundary conditions accurately predict the evolution of the flow pattern, e.g. the landmark prediction of supercritical transition to a roll pattern for the flow of water between rotating cylinders by G. I. Taylor [2] (transition due to linear instability beyond a critical rotation rate). Despite this development and the legacy of the work of Reynolds, the nature of subcritical transition (transition in the absence a linear instability) and the dynamics of pipe flow has largely remained a mystery. But much has changed following the discovery finite-amplitude solutions to the Navier–Stokes equations, for pipe flow as recently as 2003 [3]. These solutions, often referred to as ‘exact coherent states’ [4] are believed to embody the processes that sustain turbulence to and form a ‘skeleton’ for the dynamic paths taken by the evolving flow patterns. Comprehension of the nonlinear dynamics, particularly of transition in pipes, and likewise in Couette and channel flows, has progressed in leaps and bounds over the last decade, based on the study of these solutions. New more general families of solutions continue to be discovered, and their unstable manifolds are just beginning to be calculated [5–8].

The code that has evolved into Openpipeflow has played a significant role in the realisation of this odyssey. Openpipeflow offers a more simplified approach than large computational fluid

dynamics (CFD) packages – the aim during development has been to maintain a compact and readable code. Thus Openpipeflow is easily adapted for a given analysis and extendible to new numerical methods. The code has recently been upgraded with a substantially improved parallelisation, and continues to be augmented with new extensions, for example large-eddy simulation (LES).

Following the rapid expansion of computational resources that has occurred in recent times, pipe flow is a prime example of a ‘high-dimensional’ system that is receiving examination with methods previously limited to systems with only a few degrees of freedom, such as the Lorenz attractor or the Kuramoto–Sivashinsky equation; see e.g. [9,10]. In the other direction, observations from large-scale simulations of pipe flow have inspired low-order models [11,12]. Pipe flow also provides a simple setting for the development of computationally intensive new methods, such as adjoint optimisation techniques, e.g. [13].

2. Software description

Openpipeflow implements a second-order predictor–corrector scheme, with automatic time-step control, for simulation of flow on the cylindrical domain $(r, \theta, z) \in [0, 1] \times [0, 2\pi/m_p] \times [0, 2\pi/\alpha]$, where m_p and α are parameters that determine spatial periodicity. Variables in the Navier–Stokes equations are discretised in the form

$$A(r, \theta, z) = \sum_{k < |K|} \sum_{m < |M|} A_{km}(r_n) e^{i(\alpha kz + m_p m \theta)}, \quad n = 1..N, \quad (1)$$

where the points r_n are distributed on $[0, 1]$. By default the r_n are located at the roots of a Chebyshev polynomial, bunched towards the boundaries to resolve large gradients that occur in the boundary layer.² Derivatives in the radial dimension are calculated using finite differences, so that they may be evaluated using banded matrices. The number of points used, and hence the width of the bands, is an integer parameter; by default derivatives are calculated using 9 points, for which 1st/2nd order derivatives are calculated to 8th/7th order. Following the 3/2 dealiasing rule, the sums are evaluated on $3K \times 3M$ grids in z and θ respectively. Periodicity in z is a commonplace approximation that has been shown to capture all the relevant physics of turbulent flow [14] and the transition to turbulent flow [9]. The dimension θ is naturally periodic ($m_p = 1$). Rotational symmetry ($m_p = 2, 3, \dots$) is often applied, since finite-amplitude solutions typically satisfy rotational symmetry, or applied simply to reduce computational expense when the structures of interest are much smaller than the domain, e.g. near-wall vortices at large flow rates.

A pressure-Poisson equation (PPE) formulation is employed and an influence-matrix technique applied for the enforcement of boundary conditions [15]. Let \mathbf{g} be a vector of boundary conditions, written such that $\mathbf{g} = 0$ when they are satisfied. The influence-matrix technique has several nice features.:

- Alternative boundary conditions, e.g. slip or oscillations, are easily introduced by changing the single function that evaluates \mathbf{g} ;
- The usual no-slip and divergence conditions at the boundary are satisfied such that $\|\mathbf{g}\|$ is typically at the level of the machine epsilon for the given floating-point precision;
- Computational overhead is negligible compared to evaluation of non-linear terms;
- No stability issues have been observed.

² Optionally the r_n may be read in from a file, `mesh.in`. In LES simulations, for example, it may be desirable to specify the distribution of points with respect to the position of the turbulent buffer layer.

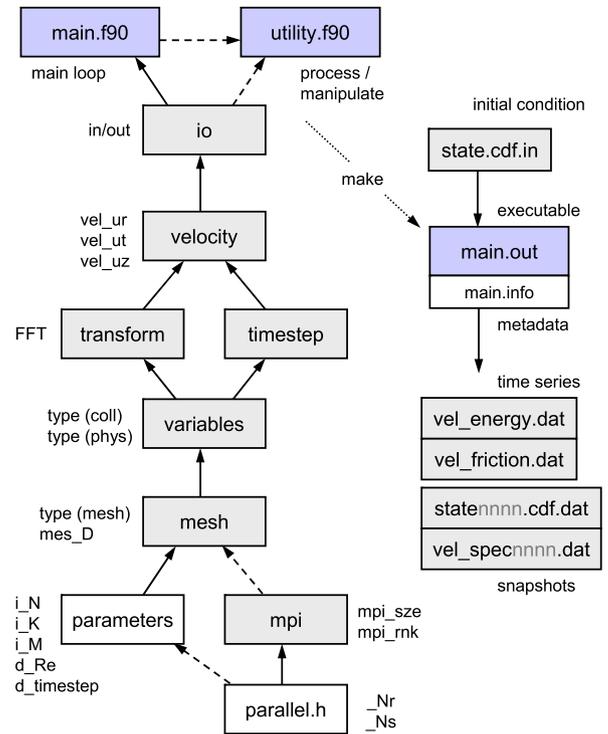


Fig. 1. Code structure and program interaction. The MPI library is not required if $_Nr = _Ns = 1$. To post-process data it is sufficient for a utility to inherit the io module. To process at run time, it is possible to inherit the whole main loop.

Utilities and templates for runtime- and post-processing are included, including a Newton–Raphson solver for the calculation and continuation of invariant solutions. The Newton solver for the pipe flow, which has a multiple-shooting option (orbits may be split into multiple sections), calls a utility that implements a combined Krylov–Trust-region approach [16]. This Newton–Krylov–Trust-region utility is designed to be integrable with any simulation code.

Openpipeflow is written in Fortran90 and uses basic modules and derived types. Esoteric extensions to the programming language have been deliberately avoided. The code makes use of FFTW, LAPACK and NetCDF libraries. Optionally, for parallel use an MPI library is required.

2.1. Software architecture and functionality

See Fig. 1 for a schematic of the code structure and program interaction. Once parameters are set and the code built, most jobs begin with a single initial condition, `state.cdf.in`. Outputs from another job, `statennnn.cdf.dat`, usually make the best initial conditions (nnnn is a 4-digit numeric label). A variety of possible initial conditions are provided in the database at openpipeflow.org. Truncation or interpolation of initial conditions with a different resolution is automatic.

A selection of utilities, plus templates for post-processing or runtime-processing, are described in the online manual (see [Appendix A](#)).

2.2. Implementation details

Linear systems that originate from the implicit solution of the viscous terms in the Navier–Stokes equations are solved using banded matrices and LU-decomposition for each Fourier mode. Nonlinear terms are evaluated pseudospectrally.

Parallelisation is achieved via a split into $_Nr$ radial and $_Ns$ axial sections, and the work is divided over $_Np = _Nr \times _Ns$

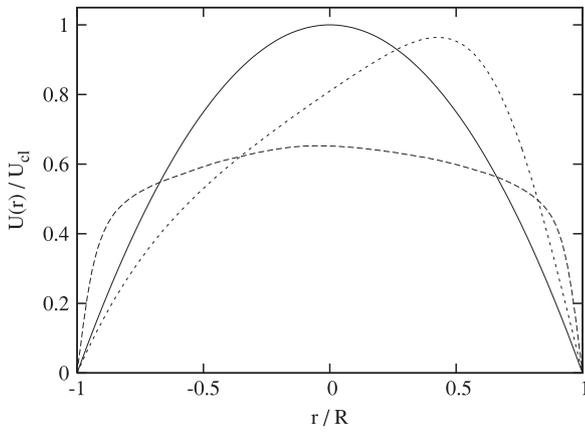


Fig. 2. Response of flow at $Re = 5300$ to a Coriolis force. Solid: Laminar flow, $E \rightarrow \infty$ (no rotation). Short-dash: Laminar flow, $E = 1$. Long-dash: Turbulent flow, $E = 1$.

cores (#-defined symbols in `parallel.h`). Due to the form of the data transposes involved in the transforms between ‘collocated’ (Fourier) and physical space (`type (coll)` and `type (phys)`), the number of cores is limited to $N \times M$. This has been a distant limitation to date.

The recent upgrade to the two-dimensional split from the one-dimensional ‘wall-normal’ split (independent 2D-FFTs) not only extends the maximum number of cores from N to $N \times M$, but also reduces the number of messages that must be sent. The transform involves two stages of FFTs and transposes, but each transpose involves only $_Nr$ or $_Ns$ cores. For a transpose involving p cores, each core must send $p - 1$ messages. Therefore, choosing $_Nr \approx _Ns \approx \sqrt{_Np}$, the number of messages is $O(2\sqrt{_Np})$ versus $O(_Np)$. This can substantially reduce time lost in latency due to the time setting up communications. Further details can be found on the Core Implementation page of the online manual (see [Appendix A](#)).

3. Illustrative examples

3.1. Modelling a Coriolis force

Does the Coriolis force, an extra force term due to rotation of Earth, affect the flow in experiments? The file `utils/Coriolis.f90` is an example utility is provided with the distribution that models this case.

The main loop of the core code already includes several calls to a null function at key points during the timestepping process; see `var_null(flag)` in `program/main.f90`. The `flag` may be used to detect the stage at which the function has been called. Here, we replace the null function with the function in `Coriolis.f90` and detect the case `flag=2`, which indicates that nonlinear terms have just been evaluated. At this point we add the Coriolis forces to the nonlinear terms. Note that no changes to the core files, including `main.f90`, are necessary.

Fig. 2 shows the mean axial flow profile for laminar and turbulent flow at an Ekman number $E = \nu/(2\Omega D^2) = 1$ for a pipe with axis oriented east–west, perpendicular to the rotation of axis for any latitude. For a pipe filled with water at 20°C , this corresponds to a diameter D of approximately 8.3 cm; $Re = 5300$ in all cases, U_{cl} is the centreline speed for laminar flow at the same mean flow rate, and $R = D/2$ is the pipe radius. For this Re , laminar flow shows a substantial response, and the profile is similar to those reported in [17]. Turbulent flow, however, shows no asymmetry. The turbulent mean profile is indiscernible from the documented test case [14].

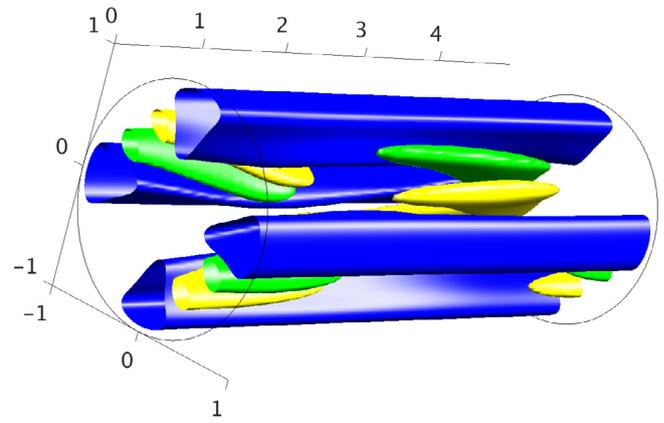


Fig. 3. N2_ML solution. (Blue) Slow ‘streaks’ – axial flow slower than the mean flow profile by >0.07 . (Yellow and green) ‘vortices’ – axial vorticity >0.2 and <-0.2 . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

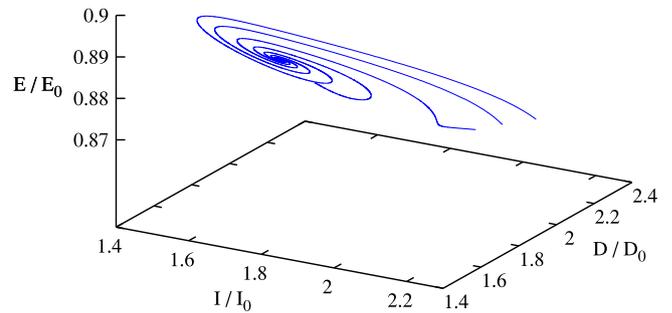


Fig. 4. Projection of the unstable manifold of the N2_ML travelling wave solution.

3.2. Unstable manifold of a travelling wave solution

A travelling wave solution is an equilibrium when considered in a frame moving at its phase speed. In this case we consider the ‘upper branch’ solution known as N2_ML, **Fig. 3**, which in its symmetry subclass has a single unstable complex eigenvalue, $0.00620 + 0.0183i(U_{cl}/R)$ (after one rotation it expands by a factor 8.4); $Re = 2400$, $\alpha = 1.25$, $m_p = 2$; see [18] for further details. For a given nearby state, the Newton–Krylov utility (`newton.f90`) can find such solutions and output their leading unstable eigenvectors (solution `state1000.cdf.dat` and real and imaginary parts of the leading eigenvector, `state1001.cdf.dat`, `state1002.cdf.dat`; available at the online Database). To visualise the unstable manifold, we use a utility (`addstates.f90`) to add small multiples ($\approx 10^{-4}$) of the real part of the eigenvector to the solution, then use these as initial conditions (`state.cdf.in`) for a set of simulations. **Fig. 4** shows a projection of the unstable manifold of N2_ML, as an outward spiral, with deformation at larger amplitudes due to nonlinearity. The coordinates are the kinetic energy E , energy input from the applied pressure gradient I , and energy dissipation D , each normalised by their respective value for laminar flow (columns of output `vel_totEID.dat`).

4. Impact and conclusions

The Openpipeflow solver aims to provide a fast but flexible code, that can be used for state-of-the-art research in the study of turbulent flows and transition.

Pipe flow is a classical setting for the development of methods for modelling and analysing dynamical systems, and Openpipeflow

has been used by several groups around the world to make an important contribution to developments in our understanding of subcritical transition, e.g. [5,7,8,11,12,19].

From these developments have arisen many new opportunities. From the theoretical viewpoint, open issues relate to comprehension of the role of newly discovered equilibria and periodic orbits. Such states are believed to provide a skeleton for the dynamics, but describing the topology of the state space for turbulence remains a challenging and active area. Pipe flow, and the study of shear flows in general, draw interest from a range of branches of mathematics and theoretical physics, e.g. pattern formation, control theory, statistical physics, experimental physics. It is an active area of cross-fertilisation for the development of mathematical and numerical methods.

From a more practical viewpoint, the dynamical systems approach is being applied in the modelling of other important flows, e.g. flows of fluids of complex rheology, e.g. stress-dependent viscosity, particulate flows and multiphase flows. The study of ‘high Reynolds number’ flows is also being influenced via application of dynamical systems techniques using LES.

Openpipeflow stands well placed to make an increasingly valuable contribution to this effort. Alongside the application of methods drawn from chaos theory, extensions to Openpipeflow have just been added for shear-thinning fluids and LES, for example. From a research perspective, plenty of exciting new developments are in the pipeline.

Acknowledgements

The author would like to acknowledge John Gibson (channelflow.org), Predrag Cvitanović (chaosbook.org), Rich Kerswell and many others for help and inspiration. The author is also very grateful for financial support from EPSRC GR/S76144/01, EP/K03636X/1 and the E.U. FP7 PIEF-GA-2008-219-233. This article was completed at the Kavli Institute for Theoretical Physics, supported in part under Grant No. NSF PHY11-25915.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.softx.2017.05.003>.

References

- [1] Reynolds O. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and the law of resistance in parallel channels. *Proc R Soc Lond Ser A Math Phys Eng Sci* 1883;174:935–82.
- [2] Taylor GI. Stability of a viscous liquid contained between two rotating cylinders. *Phil Trans R Soc A* 1923;223:289–343.
- [3] Faisst H, Eckhardt B. Traveling waves in pipe flow. *Phys Rev Lett* 2003;91:224502.
- [4] Waleffe F. Exact coherent structures in channel flow. *J Fluid Mech* 2001;435:93–102.
- [5] Pringle CCT, Duguet Y, Kerswell RR. Highly symmetric travelling waves in pipe flow. *Phil Trans R Soc A* 2009;367:457–72.
- [6] de Lozar A, Mellibovsky F, Avila M, Hof B. Edge state in pipe flow experiments. *Phys Rev Lett* 2012;108:214502. <http://dx.doi.org/10.1103/PhysRevLett.108.214502>.
- [7] Avila M, Mellibovsky F, Roland N, Hof B. Streamwise-localized solutions at the onset of turbulence in pipe flow. *Phys Rev Lett* 2013;110:224502. <http://dx.doi.org/10.1103/PhysRevLett.110.224502>.
- [8] Chantry M, Willis AP, Kerswell RR. Genesis of streamwise-localized solutions from globally periodic traveling waves in pipe flow. *Phys Rev Lett* 2014;112:164501. <http://dx.doi.org/10.1103/PhysRevLett.112.164501>.
- [9] Avila K, Moxey D, de Lozar A, Avila M, Barkley D, Hof B. The onset of turbulence in pipe flow. *Science* 2011;333:192–6. <http://dx.doi.org/10.1126/science.1203223>.
- [10] Willis AP, Short KY, Cvitanović P. Symmetry reduction in high dimensions, illustrated in a turbulent pipe. *Phys Rev E* 2016;93:022204. <http://dx.doi.org/10.1103/PhysRevE.93.022204>.
- [11] Barkley D, Song B, Mukund V, Lemoult G, Avila M, Hof B. The rise of fully turbulent flow. *Nature* 2015;526(7574):550–3.
- [12] Shih H-Y, Hsieh T-L, Goldenfeld N. Ecological collapse and the emergence of travelling waves at the onset of shear turbulence. *Nat Phys* 2015;12:245–8.
- [13] Pringle CC, Willis AP, Kerswell RR. Minimal seeds for shear flow turbulence: using nonlinear transient growth to touch the edge of chaos. *J Fluid Mech* 2012;702:415–43.
- [14] Eggels JGM, Unger F, Weiss MH, Westerweel J, Adrian RJ, Freidrich R, et al.. Fully developed turbulent pipe flow: a comparison between direct numerical simulation and experiment. *J Fluid Mech* 1994;268:175–209.
- [15] Guseva A, Willis AP, Hollerbach R, Avila M. Transition to magnetorotational turbulence in Taylor–Couette flow with imposed azimuthal magnetic field. *New J Phys* 2015;17(9):093018.
- [16] Viswanath D. Recurrent motions within plane Couette turbulence. *J Fluid Mech* 2007;580:339–58. <http://dx.doi.org/10.1017/S0022112007005459>.
- [17] Draad A, Nieuwstadt F. The Earth’s rotation and laminar pipe flow. *J Fluid Mech* 1998;361:297–308.
- [18] Willis AP, Cvitanović P, Avila M. Revealing the state space of turbulent pipe flow by symmetry reduction. *J Fluid Mech* 2013;721:514–40. <http://dx.doi.org/10.1017/jfm.2013.75>.
- [19] Willis AP, Kerswell RR. Coherent structures in localised and global pipe turbulence. *Phys Rev Lett* 2008;100:124501.