Malcolm Roberts: Research Statement

My research interests are mathematical modelling, numerical analysis, and high-performance computing. I have experience working in the areas of computational fluid mechanics, subgrid models for turbulence, and flows in complex geometries.

1 Introduction

Consider the incompressible Navier–Stokes equations. If the velocity has a high degree of regularity it may be represented efficiently via its Fourier series, which maps the quadratic nonlinear source term to a binary convolution. Truncating the Fourier series to length N, direct computation of the convolution requires $\mathcal{O}(N^2)$ operations, which can be reduced to $\mathcal{O}(N \log N)$ by transforming back to physical space. This technique, known as the pseudospectral method, is an efficient method for performing simulations that offer high accuracy and is relatively simple to implement. The technique is particularly suited for dealing with incompressible or magnetohydrodynamic flows because the Poisson equation is diagonalized in Fourier space, and thus the solenoidal constraint is easily enforced.

Pseudospectral simulations must be performed in a computational domain consisting of a periodic box, which, prima facie, greatly reduces the applicability of the method. Furthermore, dealiasing the convolution significantly increases the required memory and computational complexity of the problem. The issue of computational cost is partly addressed by making use of implicitly zero-padded convolutions (section 2), whereas one may escape the periodic box by, for example, using a penalization method (section 3) to implement complex geometries, while leaving the computational domain untouched. Using implicitly dealiased convolutions in a penalized domain allows one to perform simulations of fluids efficiently in complex domains.

Turbulent systems may be characterized by the Reynolds number R, which is the product of the characteristic length and velocity of the flow divided by the kinematic viscosity of the fluid. Kolmogorov [**?**] theorized that the number of degrees of freedom in a turbulent system depends on the the Reynolds number R as $\mathcal{O}(R^{9/4})$. Massively parallel computers and modern software libraries allow us to perform simulations with more than 10^{10} degrees of freedom [**?**]. Important physical flows may exhibit even more complexity, such as blood flow ($R \approx 3000$), the flight of a Boeing 747 ($R \approx 10^9$), and the Earth's atmosphere ($R \approx 10^{12}$), giving approximately 10^9, 10^{21}, and 10^{28} degrees of freedom, respectively. The massive size of such problems greatly exceeds the processing power of any currently available computer, and may not be realizable without a revolution in the manufacture of computer hardware [**?**]. Subgrid models of turbulence allow one to circumvent this problem by approximating or simulating the small scales. One such possibility is the method of pseudospectral reduction and the related

multispectral method, as described in section **??**.

2 Implicit Convolutions

The convolution of $f = \{f_i\}$ and $g = \{g_i\}$ is denoted $f * g$, with

$$(f * g)_i = \sum_j f_j g_{i-j}. \tag{1}$$

Convolutions are important in such areas as computer vision [**?**], statistics [**?**], signal processing [**?**], and differential equations [**?**, **?**, **?**].

For computations on real computers, the inputs f and g are of finite length N. Computing equation (**??**) directly involves $\mathcal{O}\big(N^2\big)$ operations. The convolution theorem [**?**] allows one to compute discrete convolutions using a fast Fourier transform (FFT) involving only $\mathcal{O}(N \log N)$ operations and with greatly improved accuracy over the direct method. The cyclic nature of the FFT introduces spurious modes, known as aliases, which must be removed by dealiasing the convolution [**?**, **?**]. Two common dealiasing methods are zero padding and phase-shift dealiasing, which either increases the effective problem size in each dimension by a factor of 2 or 1.5, or dealiases only a subset of the inputs. The method of implicit zero-padding [**?**] reduces the memory load by re-using work memory when the input is multi-dimensional, and has been implemented in the open-source software library `fftw++` [**?**], using `OpenMP` for multi-threading [**?**] and is applicable to convolutions involving an arbitrary number of inputs [**?**].

The method of implicit dealiasing is particularly promising for distributed memory computers using `MPI`, where the decreased memory footprint directly translates to a reduction in communication cost [**?**]. A key step towards implementing this method in a massively parallel environment is the creation of a high-performance adaptive distributed transpose algorithm [**?**], where recursive techniques were used to reduce latency when dealing with massively parallel architectures. This method has been tested and shows promise on a variety of different architectures, and I plan to extend to more general problem sizes.

Implicitly dealiased convolutions can be used on general input data. The three most important data formats are

1. non-centred complex data,
2. centred complex data with Hermitian symmetry, as used in pseudospectral simulations of nonlinear partial differential equations, and
3. non-centred real data, as used for image analysis, statistics, signal processing, and statistics on real-valued data.

Implicit dealiasing has been implemented for cases **??** and **??**. Case **??** remains to be implemented, and offers significant benefits for many real-world problems, particularly for high-dimensional data sets as found in the world of big data.

3 Penalization method

The speed and precision of the pseudospectral method are available only for certain pairs of computational domains and basis functions, for example a periodic box when using complex exponentials as basis functions. Using a Chebyshev polynomial basis allows one to escape this periodicity, but one is still restricted to rectangular domain. Another choice is to use harmonic functions tailored to the computational domain (eg cylindrical or spherical), but these are not available for general geometries, and determining them numerically can be as computationally difficult as performing the fluid simulation itself. For finite-volume or finite-element methods, for example, it may be possible to deform the grid to follow the boundary, but this is often expensive, leads to ill-conditioned linear systems, and may not be possible for moving boundaries.

These difficulties may be circumvented by using the penalty method, which leaves the computational domain relatively unconstrained and models the influence of the wall via a penalty term [**?**, **?**], which can be used to implement a wide variety of different geometries by simply specifying a characteristic function for the wall domain. A convergence proof for the Navier–Stokes equations was given in [**?**] for homogeneous Dirichlet boundary conditions. The method was implemented with homogeneous Neumann boundary conditions in [**?**]. In [**?**], we applied the penalty method to magnetohydrodynamic (MHD) flows and introduced a technique to impose non-homogeneous Dirichlet boundary conditions which do not correspond to solid-body rotation.

The extension of the penalty method to other types of boundary conditions (for example, mixed, and non-homogeneous Neumann) is an exciting direction of research, which will allow for simulations involving inflow/outflow boundary conditions and other complex physics.

The accuracy of the penalty method is governed by the penalization parameter η, with the accuracy scaling as $\mathcal{O}\left(\sqrt{\eta} \times dx\right)$ where dx is the grid spacing [**?**]. Increasing the convergence with respect to the grid spacing would greatly improve the effectiveness of this method. One possible way to achieve this is to improve the regularity of the method by matching the gradient of the fluid and the penalty field at the boundary, possibly via an iterative approach. For incompressible and MHD flows, the added restriction that the fields be divergence-free greatly complicates matters. The method for finding solenoidal penalty fields given in [**?**] may offer a method to overcome this difficulty.

The penalization parameter η imposes a time scale on the simulation which may be much smaller than that of the fluid itself, requiring simulations to be performed with very small time steps. Using implicit time-stepping techniques removes this restriction but reduces the accuracy of the penalty method. Another option is to use exponential Runge-Kutta methods [**?**], which solve the linear term exactly and has been shown to greatly reduce the cost of performing simulations of shell models of turbulence. Exponential Runge-Kutta integrators are interesting mathematical objects of their own right, and their generation involves deep algebraic results if one desires an accuracy of $\mathcal{O}\left(dt^4\right)$ or higher [**?**, **?**].

4 Shell models of turbulence

Shell models are toy models for turbulent systems; they mimic the behaviour of the Navier–Stokes or magnetohydrodynamic equations but lack the computational and analytic difficulties. They are useful for testing numerical methods [**?**], as well as theories of turbulent dynamics [**?**]. The DN, GOY [**?**], and SABRA [**?**] models are examples of such systems. In [**?**], we used these models to explore the dissipation wavenumber of highly turbulent systems with numerical simulations at very high Reynolds number contrasted with theoretical results. In [**?**], these models were used to test a numerical method which was later applied to the 2D Navier–Stokes equations in [**?**].

Shell models of turbulence can also be used to explore other aspects of fluid flow, such as the relation between intermittency and the conservation of helicity-like quantities. Moreover, by sending the shell spacing to zero, one recovers a dynamical system which can be studied analytically [**?**].

5 Self-organization and coherent structures in turbulence

Turbulence is chaotic and irregular, but it is not purely random. The trail of smoke rising from a flame is laminar close to the flame, but eventually becomes unstable, forming various flow structures known as vortices, and structures of this type have been the object of study for both the Navier–Stokes [**?**, **?**] and MHD [**?**] equations. The ability of turbulent flows to self-organize into long-lived coherent structures may allow for a deeper understanding of the underlying dynamics. Moreover, the presence of large structures suggests that the Kolmogorov estimate for the number of degrees of freedom may be overly pessimistic.

One of my projects in this area has focused on self-organization on 2D decaying turbulence, where an automated vortex census using high-resolution simulations. Using an automated vortex detection allows for better statistics over a wider range of parameters. In addition, I am interested in the self-organization of MHD flows [**?**], which may important roles in, for example, the creation of a reverse-field pinch [**?**] for tokamak fusion reactors.

6 Subgrid Models

We do not have the ideal resources to perform full-resolution direct numerical simulations (DNS); one performs simulations with the computer available, not the one that one might want or wish to have at a later time. It is tempting to simply perform simulations at low resolution, but this selectively removes the dissipation range which may produce errors at all scales of the simulation.

In order to perform useful simulations, various subgrid models have been proposed to approximate the effect of unresolved small scales, such as the k–ϵ model [**?**], large eddy simulations (LES) [**?**], and RANS-type models [**?**]. The

k–ϵ and LES models are phenomenological, and need to be tuned using DNS data which may not be available for a particular physical situation. The RANS-α model, on the other hand, dramatically modifies the dynamics at the small scales, changing physically important features such as the dissipation wavelength which may dramatically influence physical processes such as, for example, mixing and combustion.

The above-mentioned subgrid models do not simulate the small scale modes in the system, but replace their effect on the large scales with some type of model. Since there is a very large number of small-scale modes, modelling all of them is impossible, but it may be possible to perform simulations where the small scales are approximated instead of removed [**?**]. The method of spectral reduction [**?**, **?**, **?**] allows one to perform spectral simulations in which Fourier bins of averaged Fourier modes are evolved in time. Following the work on implicitly dealiased convolutions [**?**], the averaged nonlinear term has been implemented efficiently using a modified convolution theorem [**?**].

The spectrally-reduced Navier–Stokes equations produce the correct statistical-mechanical equipartition only when the bins are of equal size, which has the unfortunate effect of requiring the same approximation filter to the small and large scales. This can be averted by applying a hierarchy of differently spectrally reduced grids which are synchronized via projection/prolongation operators. This has been implemented for shell models of turbulence [**?**] and the 2D incompressible Navier–Stokes equations with periodic boundary conditions [**?**]. Multispectral reduction relies heavily on pseudospectral reduction, and one outstanding issue is whether pseudospectral reduction modifies the time scale of each grid. Application to 3D flows, and the application of more general boundary conditions are pressing avenues of research in this area.

Another approach to reducing the number of degrees of freedom in simulations is to take advantage of spatial inhomogeneity in both fully developed turbulence systems [**?**] and when parts of the flow-field are quiescent, such as in the case of a near-wall turbulence. When using uniform grids, one must increase the resolution of the simulation everywhere in order to capture the dynamics of the active region. If the active region is only a small portion of the total flow, then it would be more efficient to refine the grid only where the flow field requires it. For finite-volume or finite-element methods, this can be accomplished via adaptive multi-resolution by refining the elements locally. For spectral methods, this may be possible via a non-uniform FFT [**?**], or possibly a FFT-based convolution method specifically designed for non-uniform input. An important aspect of adaptive multi-resolution simulations is load balancing, and it would be worth considering whether FFTs can be efficiently performed on space-filling curves (eg the Hilbert curve or Z-curve).