Full parallelism of calculations of optimal flow control

1. Abstract

The project is about fully parallelizing a well-tested flow control code SEMTEX. It has been partly parallelised in a homogenous direction by Fourier transform, which is called modal parallelisation. After Fourier decomposition, 3D flow can be decomposed into a number of, weakly interacting 2D problems. There are two aims in this project. The first one is to parallelise a 2D problem by partitioning the plane and distribute the partitioned elements among the processes, which is called elemental parallelisation. The second aim is to hybrid the modal and elemental parallel techniques to realise fully 3D parallelisation. After this project, very large scale problem, e.g. flow around a cylinder at Reynolds number over one million, can be run successfully using the fully parallelised 3D DNS.

2. Introduction

The spectral/hp element method,[1] whilst being well suited to direct numerical simulation of fluid flow for many years in an academic setting, is now emerging as an attractive alternative to many traditional numerical discretisations on modern HPC hardware. As opposed to the classical finite element method, spectral/hp elements use high-order polynomial expansions on each element. On one side, this has the advantage of low dispersion and diffusion alongside exponential convergence in the polynomial order. On the other side, discretised operators are dense and have a far richer structure compared to linear expansions, meaning that they can more effectively utilise caching on modern HPC hardware. The tensor product of one-dimensional basis functions on each element also admits a rich fabric of implementation strategies.[2]

Spectral element–Fourier discretisations[3,11,2] are feasible for flows where the geometry exhibits arbitrary complexity in a sectional plane but is infinite or periodic in an orthogonal direction. In such systems, Fourier expansions can be used in the azimuthal coordinate, while arbitrary complexity can be dealt in the meridional semi-plane through use of spectral elements. In this study we are specifically interested in the case where only one coordinate direction possesses geometric homogeneity. Therefore, the 3D domain is decomposed into a sequence of spectral/hp element planes, coupled using a Fourier expansion in the third coordinate directions. SEMTEX is such an open source quadrilateral Fourier spectral element code that uses the standard nodal GLL basis functions and (optionally) Fourier expansions in a homogeneous direction to provide three-dimensional solutions for the Navier-Stokes (NS) equations.

Two typical approaches are used when parallelising this type of discretisation. One is to use mesh-decomposition in the spectral/hp element planes, and the other is to apply a modal decomposition in the Fourier direction. The latter takes advantage of the orthogonality property of the Fourier basis for linear operators. The optimal choice of parallelisation strategy typically depends on the size of the problem, the ratio of Fourier planes to spectral elements, alongside the hardware and interconnect of the parallel system. This study will hybrid these two approaches to extend the limit on the number of viable processes.

3. Fourier spectral/hp element method

Assuming the fluid to be Newtonian and the flow incompressible, the relevant equations of motion for the primitive variables (velocities, pressure), denoted as ($\boldsymbol{u}$, p), are the incompressible Navier-Stokes equations.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u},$$

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

where p is the kinematic pressure, ν is the kinematic viscosity and $\boldsymbol{u} = [u, v, w]^{\mathrm{T}}$ is the velocity.

3.1 Spatial discretisation

The SEMTEX code aims at simulating flow that is three-dimensional (3D) but homogeneous in one direction. For example, flow around an infinite turbine blade is 3D but homogeneous in the spanwise direction. Another example is flow in cylindrical coordinate where the physical variables are naturally 2π-periodic in azimuthal direction. In both cases, the velocity field can be projected exactly onto a set of two-dimensional complex Fourier modes. For simplicity, we only consider Cartesian coordinate (x, y, z) in the following, and the z direction is assumed to be homogeneous.

The velocity field can be projected onto a set of two dimensional complex Fourier modes

$$\hat{\boldsymbol{u}}_k(x, y, t) = \frac{1}{2\pi}\int_0^{2\pi} \boldsymbol{u}(x, y, z, t)\exp(-\mathrm{i}\,kz)\,\mathrm{d}\,z,$$

where k is an integer wavenumber. The velocity field can be recovered from these complex modes through Fourier series reconstruction

$$\boldsymbol{u}(x, y, z, t) = \sum_{k=-\infty}^{\infty} \hat{\boldsymbol{u}}_k(x, y, t)\exp(\mathrm{i}\,kz).$$

In practice, a finite number of modes are retained in the calculation, and the conjugate-symmetric property of the Fourier transforms of real variables is exploited, so that the negative-k modes are not required.1 Note that in cylindrical coordinate, appropriate conditions need to be applied at the axis and can be derived from solvability requirements and kinematic constraints on scalar and velocity fields at the origin.[3]

The two dimensional spectral/hp element planes span the x and y coordinate directions. The planes are discretized into quadrilateral elements and standard nodal GLL basis functions are used in each element. **Error! Bookmark not defined.** The similar approach may be equally applied when using triangular elements. Numerical integration and differential operators are constructed on a standard reference element Ω^{st} Ωst which is mapped to each Ω^e_k using a bijective map, $\chi^e\!: \Omega^{st} \to \Omega^e$, as $\boldsymbol{x} = \boldsymbol{\chi}^e(\boldsymbol{\xi})$, On each element, the solution u may be approximated as

$$u^\delta(x, y) = \sum_{n=0}^{N} \phi_n(x, y)\hat{u}_n = \sum_{p=0}^{P}\sum_{q=0}^{P} \phi_p(x)\phi_q(y)\hat{u}^e_{pq},$$

where $\hat{u}^e_{pq}$ are elemental coefficients. These correspond to the tensor-product of nodal expansion bases, $\phi_p(x)$ and $\phi_q(y)$, of order P defined as Lagrange polynomials through Gauss–Lobatto–Legendre points.

The connectivity of elements in a plane is represented by an assembly mapping A which maps the concatenated vector of elemental degrees of freedom to their global counterparts and enforces a C_0-continuity constraint. The global degrees of freedom are assembled using the relation $\hat{\mathbf{u}}^{e} = \mathbf{A}\hat{\mathbf{u}}^{g}$, where $\mathbf{A}$ is the matrix equivalent of A. This matrix is in general highly sparse and so is in practice not constructed explicitly. Operators in the spectral/hp element method are constructed elementally and applied using the sum-factorisation technique[4] as this has been demonstrated to be more efficient when operating on elements with higher-order bases.[5] The tensor-product nature of the elemental expansion bases allows matrix–vector operations to be decomposed into a sequence of smaller, more computationally efficient matrix–matrix operations, performed in each coordinate direction separately.

Taking *z* direction into consideration, the solution is expressed using a Fourier basis of $N_Z/2$ complex modes, $\phi_k(z) = e^{izk}$, to give an expansion of the three-dimensional solution on an element as

$$u^{\delta}(x,y,z) = \sum_{n} \phi_n(x,y,z)\hat{u}_n = \sum_{pqk} \phi_{pq}(x,y)\phi_k(z)\hat{u}_{pqk} \,.$$

3.2 Temporal discretisation

In the temporal discretization of the gonverning equations, a velocity-correction projection scheme is used, based on backwards differencing in time.[6] The value of a term at a new time level (*n* + 1) is explicitly extrapolated from previous steps through polynomial approximation

$$()^{(n+1)} = \sum_{q=0}^{J-1} \beta_q ()^{(n-q)} + O(\Delta t)^{J} \,,$$

while the value of derivatives at a new time level is implicitly approximated as

$$\partial_t ()^{(n+1)} = \frac{1}{\Delta t}\sum_{q=0}^{J} \alpha_q ()^{(n-q+1)} + O(\Delta t)^{J+1} \,,$$

where *J* is the integration order. In practice, at the beginning of the simulation, when *n* < *J*, n is used as the integration order. The discrete weights α_q, β_q for order up to *J* = 3 are given in literature.6

The time-step for the velocity-correction scheme commences with solution of a pressure Poisson equation following a velocity update and then the velocity is further updated using the pressure gradient. This process is formulated as

$$r\boldsymbol{u}^{*} = -\sum_{q=1}^{J} \alpha_q r\boldsymbol{u}^{(n-q)} - \Delta t \sum_{q=0}^{J-1} \beta_q rN(\boldsymbol{u}^{(n-q)}),$$

$$r\nabla^2 p^{(n+1)} = \frac{1}{\Delta t} r\nabla \cdot \boldsymbol{u}^{*},$$

$$r\boldsymbol{u}^{**} = r\boldsymbol{u}^{*} - r\nabla p^{(n+1)}\Delta t,$$

with

$$r\partial_n p^{(n+1)} = -rn \cdot \sum_{q=0}^{J-1} \beta_q (N(\boldsymbol{u}^{(n-q)}) + \frac{1}{Re}\nabla\times\nabla\times\boldsymbol{u}^{(n-q)} + \partial_t \boldsymbol{u}^{(n-q)}),$$

which is used to estimate a Neumann pressure boundary condition on boundaries where the velocity boundary conditions are of Dirichlet type.

The time-step is completed by applying a viscous correction through the solution of a Helmholtz equation (actually, a set of scalar Helmholtz equations) for $\boldsymbol{u}^{(n+1)}$ together with appropriate velocity boundary conditions at time (*n* + 1)*Δt* as

$$r\nabla^2\boldsymbol{u}^{(n+1)} - \frac{r\alpha_0 Re}{\Delta t}\boldsymbol{u}^{(n+1)} = -\frac{r\boldsymbol{u}^{**}Re}{\Delta t}.$$

3.3 Parallelisation

There are two approaches for parallel computing, one is parallel decomposition of the Fourier modes (modal parallelisation) and the other is parallel decomposition of the spectral/hp element planes (elemental parallelisation). Prior to this project, SEMTEX only enables modal parallelisation which limit parallel efficiency and scaling. The primary objective of this project is to implement elemental parallelisation and then hybrid those two approaches, i.e. combine both techniques. [7]

For modal parallelisation, the N_Z planes corresponding to $N_Z/2$ complex Fourier modes, are distributed equally among the $N_Z/2$ processes. As can be found above, after time discretization, a Helmholtz problem needs to be solved for each velocity component and the pressure at every time step. Each discrete Helmholtz problem, when discretized within a Galerkin formulation, results in a matrix problem which can be solved either directly or iteratively. Such elliptic solves are decoupled in the Fourier-transformed space and can be performed independently. The non-linear advection term is more efficiently computed in phsical space. The inverse and forward Fourier transforms, used before and after the advection calculation respectively, need to be performed and the data to be transformed must reside on the same process. In practice, this requires a transposition of the data using an MPI all-to-all operation. To support efficient differentiation in the z-coordinate direction, we additionally impose the constraint that both the real and imaginary components of each complex Fourier mode reside on the same process, since, in the Fourier space, derivatives are calculated through the multiplication $\hat{u}_k$ =a $-ik\hat{u}_k$. This restricts the maximum number of useable processes to $N_Z/2$.

In contrast, elemental parallelism partion each plane into N_{el} parts and distribute them among the processes. The partitioning of the 2D plane is implemented using the METIS library[8] and an identical partitioning and distribution amongst processes is used for each plane in the domain. The natural limit on the number of useable processes is therefore N_{el}. The nodal-graph of the mesh is partitioned among the R processes to equally distribute the number of degrees of freedom, whilst minimising the edge-cut, and therefore the inter-process communication. Elliptic solves are performed iteratively, with communication being required to exchange boundary information between adjacent elements residing on different processes at each iteration. This data exchange is implemented using the gather–scatter algorithm from Nek5000[9] which uses a global numbering of the DOFs in the plane to efficiently summate process-local contributions and distribute the result back to the participating processes.

Hybrid parallelisation combines both modal and elemental approaches by organising the available processes in a Cartesian grid,[10] see in Fig. 1. In this arrangement, the world communicator is split into a series of row communicators which support elemental parallelisation, while column communicators enable modal parallelisation. Each process belongs to precisely one row communicator and one column communicator and nominally operates on a fixed subset of elements in a fixed subset of planes. As in modal parallelism, elliptic solves are performed in Fourier-transformed space, but due to the elemental parallelism the iterative conjugate gradient solver must be used. The limit on the number of viable processes is now increased substantially to $N_{el} \times N_Z/2$. The hybrid parallelisation has been implemented in Nektar[11] and Nektar++[7] for Cartesian coordinates, whereas the difference in here is that SEMTEX supports both cylindrical and Cartesian coordinates.

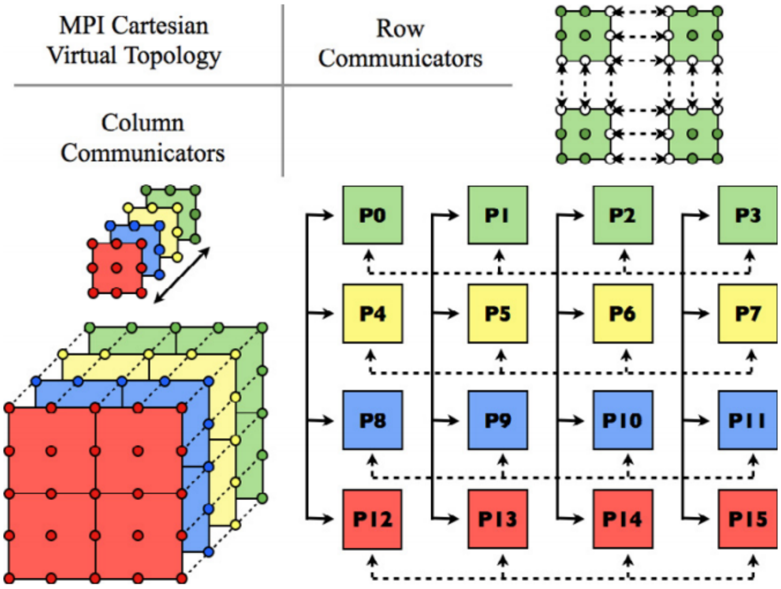

Figure 1. A simple model illustrate MPI Cartesian communicator for a hybrid parallelisation of a Fourier spectral/hp element discretisation using 4 elements per plane and 4 planes, on 16 MPI processes. Row communicators handle the communication between mesh partitions for elemental parallelisation while column communicators handle communication between planes for modal parallelisation. Figure from Ref. 7.

4. Results

Table 1 Speedup tested on ARCHER:

Cores	Wall-clock time (s) per step	Time*Cores (seconds)	Efficiency
24	5.24	125.8	1
48	2.75	132	0.95
96	1.63	156.5	0.81
192	1.1	211	0.6

The parallel efficiency has been tested on archer. Our test case is flow past a circular cylinder. The 2d plane is divided into 8577 elements and the tensor product polynomial order np=6 is used. The speedup can be found in Table 1. It can be seen that the parallel efficiency decreases with increaing core number. The efficiency is low when using 192 cores, which may be due to the large cost of communication.

5. Conclusion

In summary, we have implemented 2d parallelism in the Fourier spectral code SEMTEX. Fully 3d parallelism is achieved by hybrid the newly implemented 2d elemental parallelism with the existed Fourier (modal) parallelism. The parallel performance has been tested by simulating flow past a circular cylinder. The parallel efficiency can only be maintained for small scale calculation. The scaling still needs to be improved. The scaling can be possibly improved by introducing OpenMp and reducing communication.

References

[1]G. Karniadakis, S. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, second ed., Numer. Math. Sci. Comp., Oxford University Press, Oxford, 2005.

[2] C. Cantwell, S. Sherwin, R. Kirby, P. Kelly, From h to p efficiently: Strategy selection for operator evaluation on hexahedral and tetrahedral elements, Comput. & Fluids 43 (1) (2011) 23–28.

[3] H. M. Blackburn and S. J. Sherwin, Formulation of a Galerkin spectral element–Fourier method for three-dimensional incompressible flows in cylindrical geometries, J. Comput. Phys., 197 (2004) 759–778.

[4] S. Orszag, Spectral methods for problems in complex geometries, J. Comput. Phys. 37 (1) (1980) 70–92.

[5] P. Vos, S. Sherwin, R. Kirby, From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low-and high-order discretisations, J. Comput. Phys. 229 (13) (2010) 5161–5181.

[6] G. E. Karniadakis, M. Israeli, and S. A. Orszag. High-order splitting methods for incompressible Navier-Stokes equations. J. Comput. Phys., 97:414,1991. 138.

[7] A. Bolis, C. D. Cantwell, D. Moxey, D. Serson, S. J. Sherwin, An adaptable parallel algorithm for the direct numerical simulation of incompressible turbulent flows using a Fourier spectral/hp element method and MPI virtual topologies, Comput. Phys. Comm., 206 (2016) 17-25.

[8] G. Karypis, METIS's Manual, Department of Computer Science and Engineering, fifith ed., University of Minnesota, Minneapolis, MN 55455, March, 2013.

[9] P. Fischer, J. Lottes, D. Pointer, A. Siegel, Petascale algorithms for reactor hydrodynamics, J. Phys.: Conf. Ser. 125 (1) (2008).

[10] C. Hamman, R. Kirby, M. Berzin, Parallelization and scalability of a spectral element channel flow solver for incompressible Navier–Stokes equations, Concurrency, Pract. Exp. (2007) 1–7.

[11] C. Evangelinos, S. Sherwin, and G. Karniadakis. Parallel DNS algorithms on unstructured grids. Comput. methods in Appl. Mech. and Eng. 184.2 (2000): 401-425.