Parallel Discrete Vortex Methods on Commodity Supercomputers; an Investigation into Bluff Body Far Wake Behaviour

Kenji Takeda[*†], Owen R. Tutty[†] and Denis A. Nicole[*]

[†]Department of Aeronautics and Astronautics, and
[*]High Performance Computing Centre
Department of Electronics and Computer Science
University of Southampton SO17 1BJ, UK
ktakeda@soton.ac.uk, ort@soton.ac.uk and dan@ecs.soton.ac.uk

Abstract

Parallel discrete vortex methods are ideally suited for studying the behaviour of bluff body wakes due to their ability to capture the motion of vortex structures and lack of downstream grid boundaries. However, the availability of suitable parallel computers to run long simulations is always an issue. The convergence of the high-end workstation and commodity PC markets means that it is now possible to build cheap, powerful supercomputer-level machines at a fraction of the cost of proprietary systems. The characteristics, programming methodology and performance of such systems are discussed in this paper.

We also present results showing vortex merging behaviour in the far-wake of a circular cylinder. Previous simulations using the random walk method have shown a doubling of the shedding wavelength in the far wake compared with the near wake. New results using a deterministic vortex method are presented. The parallel vortex method used incorporates an *O(NlogN)* fast multipole method and vortex panel method to satisfy solid body boundary conditions. In order to account for viscous effects, both near the body and in the far wake, the vorticity redistribution method is used. This is considerably more accurate than stochastic methods

The aim of this paper is twofold. To present our findings on the behaviour of fully viscous, far wakes behind bluff bodies and to demonstrate that sufficient resource can be obtained on a cost-effective commodity supercomputer.

Introduction

The Discrete Vortex Method (DVM) is a technique that uses a Lagrangian framework to solve the Euler and Navier-Stokes Equations, circumventing many of the problems of grid-based methods. In this work, we use DVMs for simulating incompressible, homogenous, two-dimensional Newtonian fluid flow past bluff bodies. The main flow parameter is the vorticity of the fluid, defined as the curl of the local fluid velocity [1]. In the DVM, fluid velocity and pressure are regarded as a consequence of the solution rather than driving factors. The overall vorticity field is broken down into a number of computational elements that represent localised areas of fluid. In two dimensions these discrete vortices, also known as vortex blobs, induce a velocity on each other and are allowed to move with the fluid [1]. The vorticity, velocity and pressure fields can be calculated as a post-processing step rather than being inherently tied to the flow computation.

1 Viscous vortex method

The incompressible, Navier-Stokes equations describing fluid flow are usually written in pressure-velocity form. However, it is perhaps more natural to look at the curl of the velocity field, resulting in the vorticity form of the Navier-Stokes equations (in two dimensions),

$$\frac{\partial \omega}{\partial t} + \underline{u}.\nabla \omega = \nu.\nabla^2 \omega \tag{1}$$

$$\nabla^2 \psi = -\omega \tag{2}$$

$$\underline{u} = \nabla \times (\psi \underline{k}) \tag{3}$$

where $\underline{u}(r,t)$ is the velocity at position $\underline{r}$ and time t, $\psi(r,t)$ is the streamfunction, $\omega(r,t)$ is the vorticity, defined by $\omega \underline{k} = \nabla \times \underline{u}$ where $\underline{k}$ is the unit vector perpendicular to the plane of velocity. It is common to

operator-split the Navier-Stokes equation into convective and diffusive parts (4), which can be solved separately [4].

$$\frac{\partial \omega}{\partial t} = -(\underline{u}.\nabla)\omega \, , \; \frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega \, . \tag{4}$$

A solution to the streamfunction (ψ) is given by the convolution integral,

$$\underline{\psi}(\underline{x},t) = \int G(\underline{x} - \underline{x'})\omega(\underline{x'},t)d\underline{x'} \tag{5}$$

where G is the Poisson kernel. Letting $\underline{K} = \nabla \times G$, we can express the velocity in terms of the vorticity by means of a Biot-Savart integral such that $\underline{u} = \underline{K} * \omega$, where $*$ denotes a convolution,

$$\underline{u}(\underline{x},t) = \int \underline{K}(\underline{x} - \underline{x'})\omega(\underline{x'},t)d\underline{x'} \, . \tag{6}$$

Spatially discretising the vorticity field we obtain,

$$\omega(\underline{x},t) = \sum_{j=1}^{N} \Gamma_j \gamma \left(\left| \underline{x} - \underline{x}_j(t) \right| \right) , \tag{7}$$

where γ is the vorticity core function of the j'th discrete vortex at $r_j(t)$, Γ_j is its strength, and N the total number of discrete vortices used. Hence, we can obtain an approximation to the vortex velocity from (7),

$$\underline{u}(\underline{x},t) = \sum_{j=1}^{N} \Gamma_j \underline{K}_\delta (\underline{x} - \underline{x}_j(t)) , \tag{8}$$

where $\underline{K}_\delta = K * \gamma$. The well-tested Lamb or Guassian vortex core is used in this study (σ is the core radius),

$$\gamma(\underline{x}) = \frac{1}{\pi \sigma^2} e^{\left(-\frac{\left| x^2 \right|}{\sigma^2} \right)} . \tag{9}$$

An approach similar, in principle, to Particle Strength Exchange [12] for simulating diffusion is the vorticity redistribution method developed by Shankar and van Dommelen [17]. Circulation is distributed between vortices. The amount of circulation transferred is calculated dynamically and can be likened to a computed finite-difference formula, rather than depending on the local value of a diffusion kernel [12]. While the PSE method has been shown to give excellent results for short-time flows past bluff bodies [12, 18], we have chosen not to use it here as the vorticity redistribution method allows us to use an infinite computational domain without explicit regridding [18].

The diffusive part is solved for each vortex by calculating the fraction of vorticity to be distributed amongst other vortices within a neighbourhood, based on the typical diffusion distance,

$$h_v \equiv \sqrt{\nu \Delta t} \tag{10}$$

where the radius of the neighbourhood is defined, for the j'th vortex, as,

$$\left| \underline{x}_j - \underline{x}_i \right| \le R h_v \tag{11}$$

A maximum distance of $\sqrt{12} h_v$ is used to ensure a solution with a non-uniformly spaced set vortices [17]. The vorticity distribution is modified to satisfy the heat equation from

$$\omega^n = \sum_i \Gamma_i^n \gamma_\delta (\underline{x} - \underline{x}_i) \tag{12}$$

into

$$\omega^{n+1} = \sum_i \sum_j f_{ij}^n \Gamma_i^n \gamma_\delta (\underline{x} - \underline{x}_i) \tag{13}$$

where f_{ij} is the fraction of the $i'th$ vortex's circulation to be redistributed to the $j'th$ vortex. The values of f_{ij}^n can be found by solving a linear set of equations, which are expressed in terms of scaled relative vortex positions,

$$\zeta_{ij} = \frac{\underline{x}_j - \underline{x}_i}{h_v} \tag{14}$$

of vortices that lie within the neighbourhood radius $\zeta_{ij} \leq R$. The redistribution equations for finding the fractions can be expressed as,

$$\sum_j f_{ij}^n = 1 \tag{15}$$

$$\sum_j f_{ij}^n \zeta_{1ij} = 0; \quad \sum_j f_{ij}^n \zeta_{2ij} = 0; \tag{16}$$

$$\sum_j f_{ij}^n \zeta_{1ij}^2 = 2; \quad \sum_j f_{ij}^n \zeta_{2ij}^2 = 2; \quad \sum_j f_{ij}^n \zeta_{1ij} \zeta_{2ij} = 0. \tag{17}$$

The redistribution equations (15) - (17) have physical significance, as they express conservation of circulation, centre of vorticity, linear and angular momenta. Additional, higher-order moment equations can be used for increasing the order of accuracy arbitrarily, from $O(h_v)$ to $O(h_v^m)$ [17]. These equations can be solved using standard linear programming techniques, such as the Revised Simplex algorithm, for each vortex. The additional positivity constraint,

$$f_{ij}^n \geq 0, \tag{18}$$

ensures that no reverse vorticity is introduced and is necessary as a stability condition, preventing the L_1 norm $\Gamma^n = \sum_i \left| \Gamma_i^n \right|$ of the circulation from growing. If no feasible solution exists, new vortices are added until one can be found. This helps to stabilise the convective calculation by acting as a dynamic regridding step that inserts new vortices in a consistent manner where there are significant gaps in the vorticity field. In particular, the centre of vorticity condition, equation (16), may not be satisfied where vortices are unevenly distributed about the vortex whose circulation is being redistributed. While this increases the number of vortices in the computation, it is a judicious algorithm that only adds vortices where necessary. This ensures that a good representation of the vorticity field is maintained and is an intrinsic part of the diffusion calculation. Numerical experiments show that problem size grows approximately linearly with time when a suitable vortex circulation cut-off strength is applied [17, 20, 21]. By optimising the Revised Simplex routine, we have reduced the CPU time spent performing the redistribution step to typically around 10% of the velocity summation time for the simulations presented in section 4.

2   Commodity supercomputing

In order to perform long-time simulations using the discrete vortex method with large numbers of particles it becomes necessary to run on parallel supercomputers to obtain results in a reasonable time. However, access to such facilities is often restricted due to their high capital and running costs, with centralised administration and large numbers of users competing for time being the norm. These systems provide good performance for fine, medium and coarse-grained problems, but rapid advances in processor and network technologies mean that it is becoming possible to build supercomputer-level systems from commodity hardware (and software) which run medium- and coarse-grained parallel codes well.

Moore's law states that computing power approximately doubles every eighteen months. The main driver for this is the desktop PC market, in which volumes of sales are orders of magnitude greater than the workstation and supercomputer markets. Until recently good floating point performance has not been a characteristic of desktop PC central processing units (CPUs), precluding them from the majority of scientific applications. However, introduction of chip architecture features such as superscalar cores and multiple instruction pipelining mean that their performance on numerical codes is respectable [8]. Convergence of workstation and PC markets has induced an aggressive pricing between major vendors, and there is now little distinction between the top-end PC's and workstation systems.

Standardisation of peripherals such as memory, hard disk drives and networking means that using Mass Market, Commodity, Off-the Shelf (M^2COTS) technologies to build networks of workstations (*NOW's*) or Piles of PC's (*POPC's*) is viable in the long-term. The NASA Beowolf project [2] has pioneered the development of M^2COTS systems. These are based on Intel Pentium and Digital Alpha processors, running the Linux, POSIX-based, operating system. Connectivity is achieved via fast Ethernet, although some developers have chosen higher capacity interconnects such as Myrinet [14]. The possibility of using the Microsoft Windows NT operating system can also be effective, and is particularly attractive due to the availability of good, globally optimising compilers at economic prices [8].

While the per-node processing power on the desktop is of the same order as typical supercomputer systems, the differentiating factor is in the communications infrastructure. Proprietary systems such as the Cray T3E and Hitachi SR2201 machines have high bandwidth, low latency interconnects that enable fine-grained parallel applications to run efficiently. However, commodity systems based on fast Ethernet are limited to the performance of the hardware, around 11 Mbytes/sec peak bandwidth. In addition to this limitation, the overheads of the operating system and communications software affect performance further. Typical asymptotic bandwidth and latencies of commodity and proprietary supercomputer interconnects are shown in Table 1 using a unidirectional Message Passing Interface *pingpong* benchmark [9].

Interconnect	Latency (μsecs)	Asymptotic Bandwidth (Mbytes/sec)
Fast Ethernet (Linux/MPICH)	170	8.6
Myrinet (Linux/FM)	32	72
SGI Origin 2000	13	120
SGI/Cray T3E-900	12	180
Meiko CS2	113	33
IBM SP2 (Thin 2)	52	34

Table 1. Communications performance of commodity and proprietary memory interconnects using unidirectional MPI pingpong *benchmark.*

As can be seen from Table 1, commodity networking hardware is slower than proprietary systems. However, this level of performance is acceptable for a variety of applications, and providing the application developer takes this into account it is possible to utilise these systems effectively. The proprietary Myrinet interconnect is a PCI-based add-on card for PC's and workstations capable of good message passing performance. However, the price of such specialist peripheral hardware is considerable, £1000 (GB) per node at Q2 1998 prices. At the software level, Illinois Fast Messages [14] provides an efficient communications paradigm on which protocols such as MPI can be built utilising the Myrinet hardware.

Due to their relatively poor communications performance, commodity systems (i.e. with commodity networking) must be programmed carefully so that this drawback does not become an intractable bottleneck in any parallel application. CFD codes are often tightly coupled and therefore require more care when porting to run on commodity systems. While the latency of fast Ethernet is high, bandwidth is acceptable for large message sizes. This has been taken into account in the implementation of the parallel vortex method used here, and is discussed in more detail in section 3.

3   Parallel implementation

This research is a continuation of the work carried out by Clarke and Tutty [6] to construct a parallel discrete vortex method for viscous flow simulation. A singularity-free vortex panel method is used to satisfy local body boundary conditions [6]. An $O(NlogN)$ fast multipole algorithm based on van Dommelen and Rundenstiener [24] is implemented which uses an orthogonal, recursive bisection zonal decomposition strategy to generate the zonal structure [6]. A second-order Runge-Kutta time-stepping scheme is used to solve the convective step. Vorticity redistribution is performed once per timestep, although accuracy could be arbitrarily improved by incorporating the redistribution method inside the multi-step convective scheme.

The grid-free nature of the discrete vortex method makes it challenging to map efficiently onto a distributed memory parallel computer. A domain decomposition approach is used to split the computational domain up evenly between processors, with approximately equal numbers of vortices on each processor. The particles are spread evenly at every time step to ensure good load balancing. A *master* processor is used to handle the vortex panels and time step new vortices created at the boundary. This also acts as host processor, handling all file I/O. The flowfield is divided in the flow direction with equal numbers of particles on each of the remaining *worker* processors. This arrangement is shown in Figure 1. To perform the certain collective communications excluding the master processor a separate *worker* MPI communications group is created with the master processor being assigned rank *P-1* (not rank *0*, as is usual), *P* being the total number of processors used.
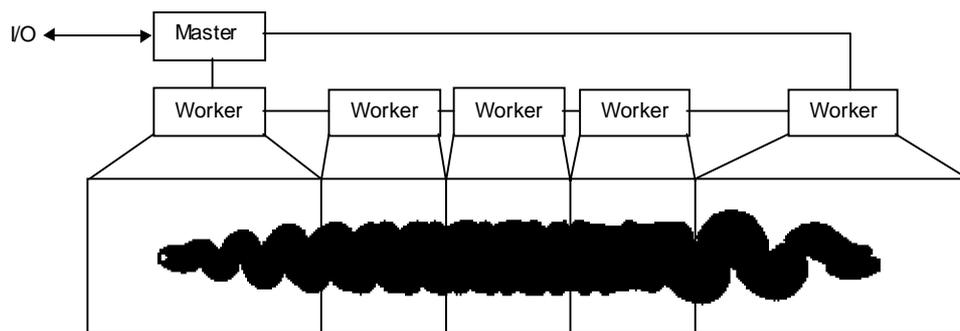

Figure 1. Parallel implementation, using task parallelism and domain decomposition.

The computationally intensive parts of the code are the zonal decomposition, velocity summation and vorticity redistribution phases. The zonal decomposition and velocity summation algorithms have been parallelised using a domain decomposition approach with systolic array communications which proceeds as follows: Each processor is assigned *N/(P-1)* vortices divided in the streamwise direction, with the last worker picking up the extra *N mod (P-1)* vortices. The zonal decomposition is then performed on each processor's local set of vortices. The velocity induced on each local vortex by the remaining local vortices is then calculated. The zonal data and local vortex positions and strengths are passed anti-clockwise around the loop. The effect of these remote vortices on the local set are then calculated. This process is repeated *P* times, so that each processor ends up with its own zonal decomposition data that is reused for the second Runge-Kutta substep. This is possible as vortices that are initially grouped together remain so when a small time step is used [6].

The vorticity redistribution method uses a replicated data approach to ensure consistency across processor boundaries. While the domain decomposition determines which vortices to calculate redistribution fractions for, vortices from all other processors are included in the redistribution calculation when necessary. This is done so that vortices more than one processor domain away from the redistribution vortex can be included easily. This situation may occur in areas where there are many particles clustered together in the streamwise direction, when large numbers of processors are used, and when the wake length is short.

To make any parallel program scale efficiently, the ratio of computation to communication must be high. However, proprietary systems typically offer low latencies and high bandwidth (see Table 1), allowing the

developer some freedom in implementing algorithms in parallel. Commodity supercomputer systems do not offer such fast communication performance, so care must be taken to minimise this bottleneck.

Reducing necessary communications is the obvious way of obtaining good scalability. When data must be exchanged between processors, it is desirable to minimise not only the amount of data sent but also the number of messages. In this way it is possible to amortise the time lost due to the start-up overhead (latency) of each message against the bandwidth. This can be achieved by using buffering, amalgamating several messages into one. This has a memory penalty, but this has been avoided by recycling arrays. Temporary working arrays allocated for calculation phases are reused as message buffers.

Custom packing routines have been used in this parallel application, as it has been shown that those supplied with MPI and other communications libraries can be surprisingly inefficient [9]. This is due to the efficiency of the particular implementation, and the underlying hardware. Custom-packing routines have been shown to be up to three times faster than standard (MPI, PVM) routines on some machines, while on others the efficiency is comparable [9]. It is particularly important to use buffering for global, collective communication operations. Implementations of these routines, such as *MPI_Gather* and *MPI_Reduce*, are in the worst case sets of blocking send and receive calls that cycle through each processor on a machine sequentially. The penalty for invoking many short collective operations can therefore be especially severe.

The advantages of running on parallel platforms can yield substantial reductions in the time for a given simulation. Application scalability is critical to make most effective use of the available resource and to justify the additional effort required to parallelise the code. The most common measure of scalability is the parallel *speedup*. There are various definitions, the most commonly accepted being,

$$speedup = \frac{run\ time\ on\ single\ processor}{run\ time\ on\ N\ processsors} \qquad (19)$$

The single processor run time should be for a sequential version with no communication overheads. In this case we measure the speedup relative to timings on two processors as we use a master/slave approach. The total run time is perhaps the most directly useful measure of how effective a parallel implementation is, as job turnaround time is ultimately what the end-user is concerned with. For these measurements a simulation of flow past an impulsively started cylinder at Reynolds number 5000 was run with 100 boundary panels and a time step of 0.10. The simulation was run to t=2.50, with restart file dumps every ten time steps.

Figure 2 shows performance on a 16-processor PC cluster running Windows NT. This system consists of 16 Intel Pentium II 300MHz PC's, each with 128 Mbytes RAM and connected by fast Ethernet and Myrinet. The code was compiled using Microsoft Visual C++ version 5.0 with default optimisations under Windows NT version 4.0. The Illinois Fast Messages implementation of MPI was used [14]. The code has also been benchmarked on a Meiko CS2 proprietary supercomputer with 16, 125MHz UltraSparc II processors with 64 Mbytes RAM each. Speedup figures are shown in Table 2.

Parallel scalability on moderate numbers of processors is good. This is particularly pleasing for the clusters running fast Ethernet networking, as it shows robustness against the relatively slow communications performance of these systems. It is interesting to note that the performance on the Pentium II cluster using fast Ethernet and Myrinet is very similar, given their raw communications performance differs by almost an order of magnitude (Table 1), as does their cost. Performance of the commodity systems is comparable to that of the Meiko CS2. In terms of price/performance it shows how clusters can be harnessed for CFD calculations, although the peculiarities of the machine architecture must be taken into account. The relatively poor scalability on 16 processors is due to there only being about 1000 vortices per processor by the end of this benchmark run.

Number of	relative speedup		
processors	Fast Ethernet	Myrinet	CS2
2	1	1	1
4	2.16	2.21	2.39

| 8 | 4.75 | 5.19 | 4.91 |
| 16 | 6.02 | Not run | not run |

Table 2. Relative speedup on Pentium II cluster using fast Ethernet and Myrinet networking, and Meiko CS2 proprietary parallel computer
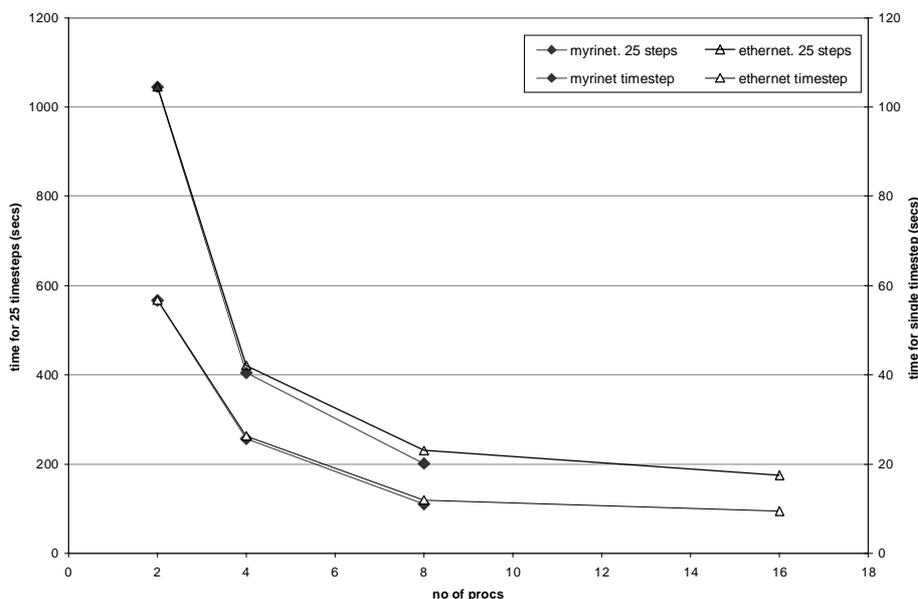

Figure 2 shows the elapsed run time of the vortex code on the Pentium II cluster using fast Ethernet and Myrinet. The cumulative timings for the first 25 time steps are shown, together with the time taken for the 25th time step.

4 Far wake behaviour

The flow behind a circular cylinder is a widely studied problem in fluid mechanics, with the ubiquitous Von Karman vortex street being one of the most recognisable coherent structures in unsteady fluid flow. In this section we investigate the long-time evolution of a cylinder wake in a purely two-dimensional, homogenous, incompressible Newtonian fluid using the discrete vortex method.

The processes of two-dimensional flow are known to be quite different from those in three-dimensions. A reverse energy cascade, from small scales to large scales is known to exist [13]. This has been an area of significant interest to meteorologists and oceanographers. This is due to the pseudo, two-dimensional nature of the atmosphere and oceans, in which stratification and rotational effects constrain fluid motion.

It has been shown experimentally [7] and numerically [23] that cylinder wakes can become unstable without any additional forcing. Vortex merging and formation of couples (pairs of vortices with oppositely-signed strength) has been observed [7, 11, 23] for Reynolds numbers of $O(10^2)$ to $O(10^3)$.

Previous computations by Tutty and Clarke [6, 23] using a hybrid diffusion velocity/random walk vortex method reveal such merging and pairing behaviour. In this study, we perform simulations of flows past circular cylinders at *low* (≤150) and *moderate* (≤5000) Reynolds numbers using a vorticity redistribution method. For a number of test problems, for bounded and unbounded flows, this deterministic method has been shown to be more accurate than the diffusion velocity and random walk methods [17, 20, 21]. Its intrinsic dynamic regridding step ensures that there is good resolution throughout the cylinder wake, and in any vortical structures that may move outside of this region.

Test cases for flows at Reynolds number 100, 150 and 1000 are presented; where Reynolds number and time (τ) are normalised with respect to cylinder diameter. All test cases were run using relatively coarse

parameters to allow long-time runs to be carried out. 100 vortex panels were used and a time step of 0.10. The vortex panels used have a parabolic approximation to the surface shape and an overlapping linear vorticity distribution so that the representation is singularity-free [6]. Multiple vortices are created along each panel, thereby increasing resolution locally, with the core size based on a fraction of the panel length. A dynamic vortex generation scheme is implemented on the panels in which a maximum vortex strength criterion is applied. If a vortex exceeds this maximum then it is split into several weaker vortices spread along the panel. This helps to ensure that sufficient numbers of particles are generated to particle maintain overlap. The growth in problem size is linear with time, with the rate being dependent on the number of panels used, time step and Reynolds number.

Extensive testing shows that these numerical parameters appear to resolve the physical flow features at these Reynolds numbers [20]. The validation process included simulations of the short-time flow past an impulsively started cylinder using drag evolution, wake geometry, vorticity and streamline plots to compare our scheme with experiment and other computations [12, 16, 19, 25]. The agreement is surprisingly good given the coarse parameters used. Vorticity contour plots are used here to visualise the cylinder wake, as they reveal the underlying structure and vortex motion clearly.

The characteristic signature for flow past an impulsively started cylinder at Reynolds number of 100 is the von Karman vortex street. Experiments can be performed in bulk fluid and retain two-dimensionality if performed carefully [25]. Our computations agree with experimental and computational results [10, 11, 15, 25] to within experimental and computational error. Taneda observed [22] in the range 60<Re<150 that the primary vortex street breaks down approximately 50 diameters downstream, re-arranging itself into a secondary vortex street further downstream with a longer wavelength. The exact ratio of the wavelengths of the primary to secondary vortex streets, denoted by a_1 and a_2 respectively, is an area of conjecture [22] but is known to be approximately two.

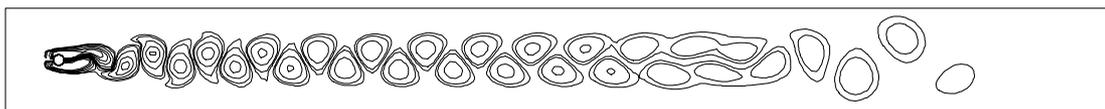*Figure 3.Vorticity contours for Re=100 at τ=100.0 using the vorticity redistribution method.*

The dynamics of the merging region are of key interest. The reason for the breakdown of the primary vortex street is not fully understood [11], and authors have used inviscid vorticity dynamics to try and explain the breakdown for idealised rows of Gaussian vortices. In our fully viscous simulation we can see how this merging region behaves with respect to convection and diffusion. Results for Reynolds number 100 are shown in Figure 3

Initially vortex pairs are shed and remain distinct, interacting but not coalescing as their separation distance is too large. As they diffuse, however, they begin to deform, elongating in the streamwise direction as regions of oppositely-signed vorticity come into contact. This elongation eventually causes like-signed vortex patches to merge. This process continues until convective effects cause vortices to roll-up from alternate sides. This severs the opposite vortex, causing it to be shed, forming the secondary vortex street. Throughout this process at the tail-end of the merging region, vortices nearer the cylinder continue to elongate forming a near-parallel shear flow. This acts to draw in vortices, ensuring that the merging region persists as vortices are shed from its tail.

These processes are consistent with the experimental and numerical evidence presented by Karasudani and Funakoshi [11]. The wavelength ratio, a_2/a_1, for our simulation is approximately 1.9 compared with a range of 1.9 to 2.3 from experiment [11].

The processes for merging and formation of the secondary vortex street at Reynolds number 150 (Figure 4) are similar to that occurring at Reynolds number 100 with some notable differences. Start-up effects are more pronounced in this higher Reynolds number flow. The formation of a strong initial vortex couple is not seen at the lower Reynolds number, but is characteristic of more convective flows such as at Reynolds number 1000 discussed below. As Strouhal frequency increases with Reynolds number, more vortices are

shed from the cylinder in a given time interval. As the wavelength of the primary vortex street is shorter at this Reynolds number, merging occurs earlier than at Reynolds number 100. The net effect is that the merging region is also longer. At τ=125.00 the end of the merging region is a continuous parallel shear flow, with no distinguishable vortex centres.

Measurement of the secondary vortex street wavelength is difficult as a the regular pattern is yet to be established. However, our results support the experimental data of Karasudani and Funakoshi [11] and Taneda [22], that show a decrease in secondary to primary vortex street wavelength with increasing Reynolds number, in the range 1.7 to 1.9.

Figure 4.Vorticity contours for Re=150 at τ=125.0 using the vorticity redistribution method.

It is known that the wake behind a circular cylinder becomes three-dimensional above a critical Reynolds number of approximately 189 [25]. In order to study truly two-dimensional flow dynamics above Reynolds numbers of 189, special techniques must therefore be used. Soap films are one such method [7], providing a thin layer of fluid in which experiments can be performed. Results from soap-film experiments performed by Couder and Basdevant [7] show the formation of distinct vortex couples at Reynolds numbers above approximately 400. We present results at Reynolds number 1000 using a vorticity redistribution method (Figure 5).
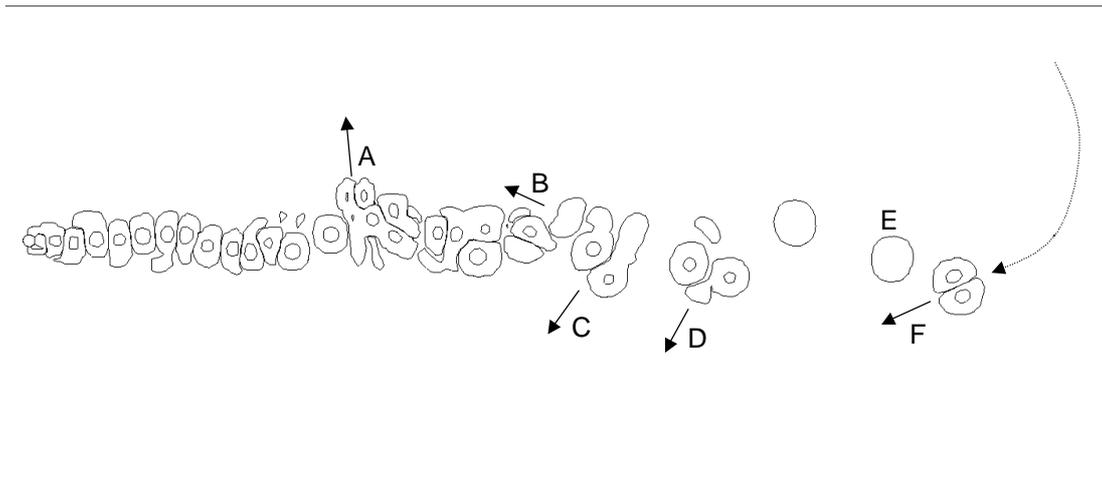

Figure 5.Vorticity contours for Re=1000 at τ=86.40 using the vorticity redistribution method.

Figure 6.Discrete vortex particle positions for Re=1000 at τ=50.00 using a hybrid diffusion

velocity/random walk method.

Figure 6 shows a result of Tutty and Clarke [23] for a Reynolds number of 1000 at τ=50.00 using a hybrid diffusion velocity and random walk method [6]. Vortex couples, many of which are moving upstream, are apparent in both simulations. In Figure 5 vortex couples labelled B, C, D and F are all moving upstream. Plots at earlier times reveal that vortex couple F is due to startup effects, and is ejected above the vortex street overtaking the startup vortex (E) before re-crossing the wake axis. Complex merging behaviour is occurring and there appears to be some compression of the near wake due to couples B and C. This seems to be the cause of couple A being ejected from the vortex street, and there being a slight tilt to the shed vortex pairs. One might expect the random walk to induce some chaotic motion to the solution and this is evident, but phenomenological similarities with results using the vorticity redistribution method are apparent. Shorter runs at Reynolds number 550 also show striking similarities between diffusion velocity/random walk and vorticity redistribution simulations in terms of vortex merging and couple formation. This supports the idea that the behaviour seen in previous computations [23] are physical, and not numerical artefacts, and are characteristic, two-dimensional phenomena driven by instability of the flow.

4   Conclusions

In this paper we have discussed the characteristics of commodity supercomputers and their application. In particular, amortising message latency against bandwidth enables good performance to be realised on these systems. Implementation of a parallel discrete vortex method based on the vorticity redistribution method has been described, and good speedup reported on a variety of parallel platforms.

Simulations of two-dimensional flow past a circular cylinder at Reynolds numbers of 100 and 150 reveal vortex merging mechanisms that lead to the breakdown of the primary wake and formation of a secondary wake of increased wavelength. At Reynolds number 1000, vortex pairing behaviour and destabilisation of the cylinder wake is apparent. This is similar to previous results using a hybrid random walk/diffusion velocity method, but the current technique is purely deterministic which suggests that this type of behaviour is physically significant.

Acknowledgements

References

1   BATCHELOR, G.K., 1967. *An Introduction to Fluid Dynamics*, Cambridge University Press.

2   BECKER, D. J., STERLING, T., SAVARESE, D., DORBAND, J. E., RANAWAK, U. A., PACKER, C. V.,"Beowolf: A Parallel Workstation for Scientific Computation", *Proc. International Conference on Parallel Processing 1995*.

3   BOUARD, R. and COUTANCEAU, M., 1980. "The Early Stage Development of the Wake Behind an Impulsively Started Cylinder for $40 < Re < 10^{4}$", *J. Fluid Mech*., **101**, p. 583.

4   CHORIN, A.J. "Numerical Study of Slightly Viscous Flow", *J. Fluid. Mech*., **57**, (1973), pp. 785-796.

5   COX, S. J., NICOLE, D. A. and TAKEDA, K., "Commodity High Performance Computing at Commodity Prices", *Proc. 21st World Occam and Transputer User Group Technical Meeting*, (1998).

6   CLARKE, N. R. and TUTTY, O. R., "Construction and Validation of a Discrete Vortex Method for Two-dimensional Incompressible Navier-Stokes Equations", *Comp Fluids*, **23**, No. 6, (1994), pp. 751-783.

7   COUDER Y. and BASDEVANT C., 1986. "Experimental and Numerical Study of Vortex Couples in Two-Dimensional Flows", *J. Fluid Mech*., **173**, (1986), pp. 225-251.

8   EMERSON, D., MAGUIRE, K., TAKEDA, K and NICOLE, D. A., 1998. "An Evaluation of Commodity Supercomputers for CFD Applications", *Proc. Parallel CFD '98, Taiwan, 11-14 May, 1998*.

9 GETOV, V., HEY, A. J. and HOCKNEY, R., "Performance Analysis, Evaluation and Optimisation Tutorial", *4*[th] *Intl. Europar Conference*, *Southampton, UK, 1998.*

10 KARNIADAKIS, G. E. AND TRIANTAFYLLOU, G. D., "Frequency Selection and Asymptotic States in a Laminar Wake", *J. Fluid Mech*, **199**, (1989).

11 KARASUDANI, T. and FUNAKOSHI, M., "Evolution of a Vortex Street in the Far Wake of a Cylinder", *Fluid Dynamics Research*, **14**, (1994), pp. 331-352.

12 KOUMOUTSAKOS, P. and LEONARD, A., "High Resolution Simulations of the Flow Around an Impulsively Started Cylinder using Vortex Methods", *J. Fluid Mech*, **296**, (1995), pp. 1-38.

13 MCWILLIAMS, J. C., "The Emergence of Isolated Coherent Vortices in Turbulent Flow", *J. Fluid Mech.*, **146**, (1984), pp. 21-43.

14 PAKIN, S., KARAMCHETI, V. and CHIEN, A. A., "Fast Messages (FM): Efficient Portable Communication for Workstation Clusters and Massively-Parallel Processors", *IEEE Concurrency*, **5**, no. 2, (1997), pp. 60-73.

15 ROSHKO, A., "On the Development of Turbulent Wakes from Vortex Streets", *Natl Advisory Committee Aeronaut., Rep.* 1191 (1954)

16 SAIKI, E. M. and BIRINGEN, S., "Numerical Simulation of a Cylinder in Uniform Flow; Application of a Virtual Boundary Method", *J. Comp. Phys.*, **123**, (1996), pp. 450-465.

17 SHANKAR, S. and VAN DOMMELEN, L., "A New Diffusion Procedure for Vortex Methods", *J. Comp. Phys.*, **127**, (1996), pp. 88-109.

18 SHIELS, D., "Simulation of Controlled Bluff Body Flow with a Viscous Vortex Method", PhD Thesis, California Institute of Technology, USA, (1998).

19 TA PHUOC LOC and BOUARD, R, "Numerical Solution of the Early Stage of the Unsteady Flow around a Circular Cylinder: a Comparison with Experimental Visualisation and Measurements", *J. Fluid Mech.*, **160**, (1985), pp. 93-117.

20 TAKEDA, K., "Parallel Discrete Vortex Methods for Viscous Flow Simulation", PhD Thesis, University of Southampton, UK, (1999).

21 TAKEDA, K., TUTTY, O. R. and FITT, A. D., "A Comparison of Four Viscous Models for the Discrete Vortex Method", *Paper number 97-1977, 13*[th] *AIAA CFD Conference, Snowmass, Colorado, 29 June - 3 July, 1997.*

22 TANEDA, S., "Downstream Development of Wakes Behind Cylinders", *J. Phys. Soc. Japan,* **14**, (1959), pp. 843-848.

23 TUTTY, O. R. and CLARKE, N. R., "Long-time Flow Past a Circular Cylinder", *presented at XIXth Intl. Congress of Theoretical and Applied Mech., Kyoto, Japan, August 25-31, 1996.*

24 VAN DOMMELEN, L. and RUNDENSTEINER, E. A., "Fast, adaptive summation of point forces in the two-dimensional Poisson equation", *J. Comp. Phys.*, **83**, (1989), pp. 126-147.

25 WILLIAMSON, C. H. K., "Vortex Dynamics in the Cylinder Wake", *Ann. Rev. Fluid Mech.*, **28**, (1996)