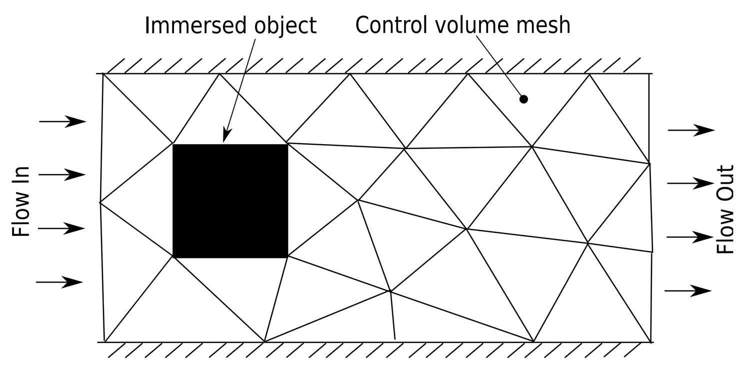
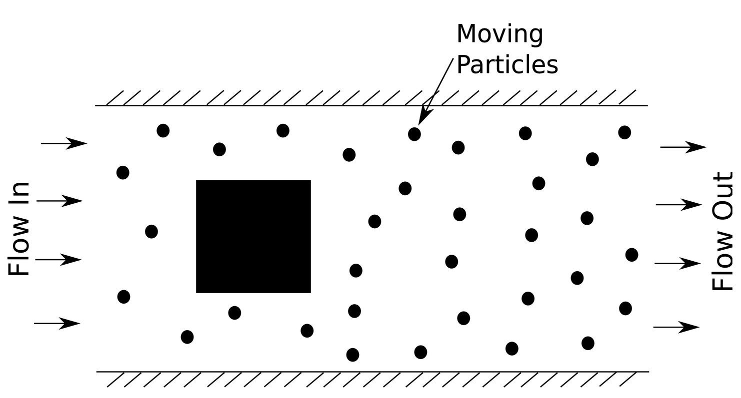
Acceleration of mesh-free CFD using CUDA

Ruairi M. Nestor[1], Gilles Civario[1], Libor Lobovsky[2], Nathan J. Quinlan[2]

1: Irish Centre for High-End Computing, The Tower, Trinity Technology & Enterprise Campus, Grand Canal Quay, Dublin 2, Ireland
2: Mechanical and Biomedical Engineering, College of Engineering & Informatics, National University of Ireland, Galway, Ireland



Traditional Mesh-Based CFD Method

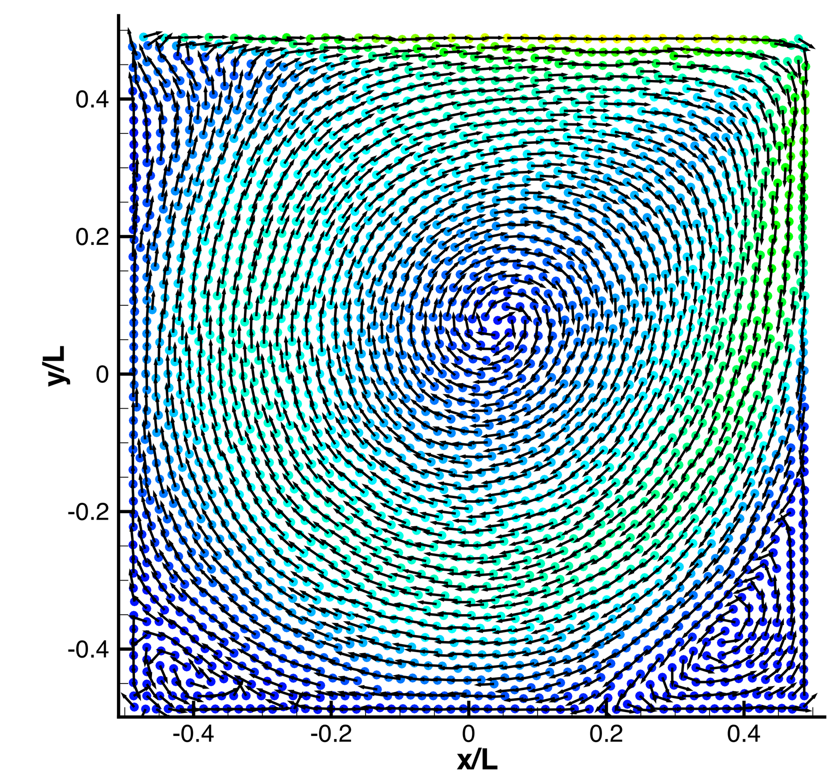


Mesh-Free CFD Method

The Finite Volume Particle Method

Computational Fluid Dynamics (CFD) is a numerical flow analysis technique. Traditional CFD methods rely on a fixed mesh of control volumes to represent the fluid. Mesh-based methods can be problematic for problems with moving walls or interfaces because special procedures are needed to deform the mesh.

The Finite Volume Particle Method (FVPM) [1] is a novel mesh-free method for Computational Fluid Dynamics (CFD). In mesh-free CFD, the fluid is represented by a set of moving particles. Mesh-free methods are better suited to problems with moving wall boundary conditions and interfaces than traditional mesh-based methods.



Lid-driven cavity flow computed using the FVPM. This type of problem is a commonly used benchmark for CFD algorithms.

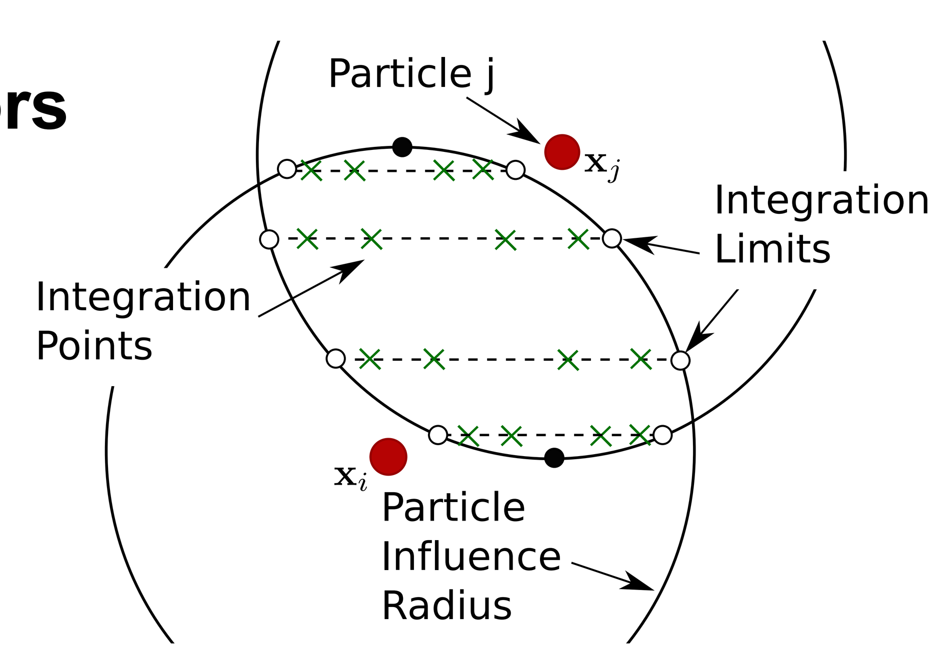
CUDA-accelerated components

A few key components of the FVPM code account for the majority of the computational effort. To date, the components highlighted in blue in the following table have been accelerated using CUDA:

FVPM Component	% Total CPU Time Serial
Particle interaction vectors	87.4
Gradients of flow variables	3.7
Flow variable rates of change	3.5
Particle volumes	2.0

Particle Interaction Vectors

The most computationally demanding part of the FVPM algorithm is the computation of the particle interaction vectors. These vectors are typically computed using numerical integration in the overlap region between each particle pair.



The interaction vector for each particle pair can be computed independently of all others. Therefore, each interaction vector is computed by a single CUDA thread.

```
//Original Serial Looping Structure
for(i=0; i < number_of_particles; i++){
  for(j=0; j < neighbours_of_i; j++){
    for(k = 0; k < number_of_integration_points; k++){
      interaction_vector[i][j] += …;
    }
  }
}
```

Each CUDA thread is assigned a loop over the integration points

Gradients of Flow Variables

To improve the spatial accuracy of the the FVPM, it is necessary to reconstruct the flow variables to the interfaces between each particle pair [2]. This is performed by computing the gradient of each flow variable at each particle on the basis of a Smoothed Particle Hydrodynamics approximation [3].

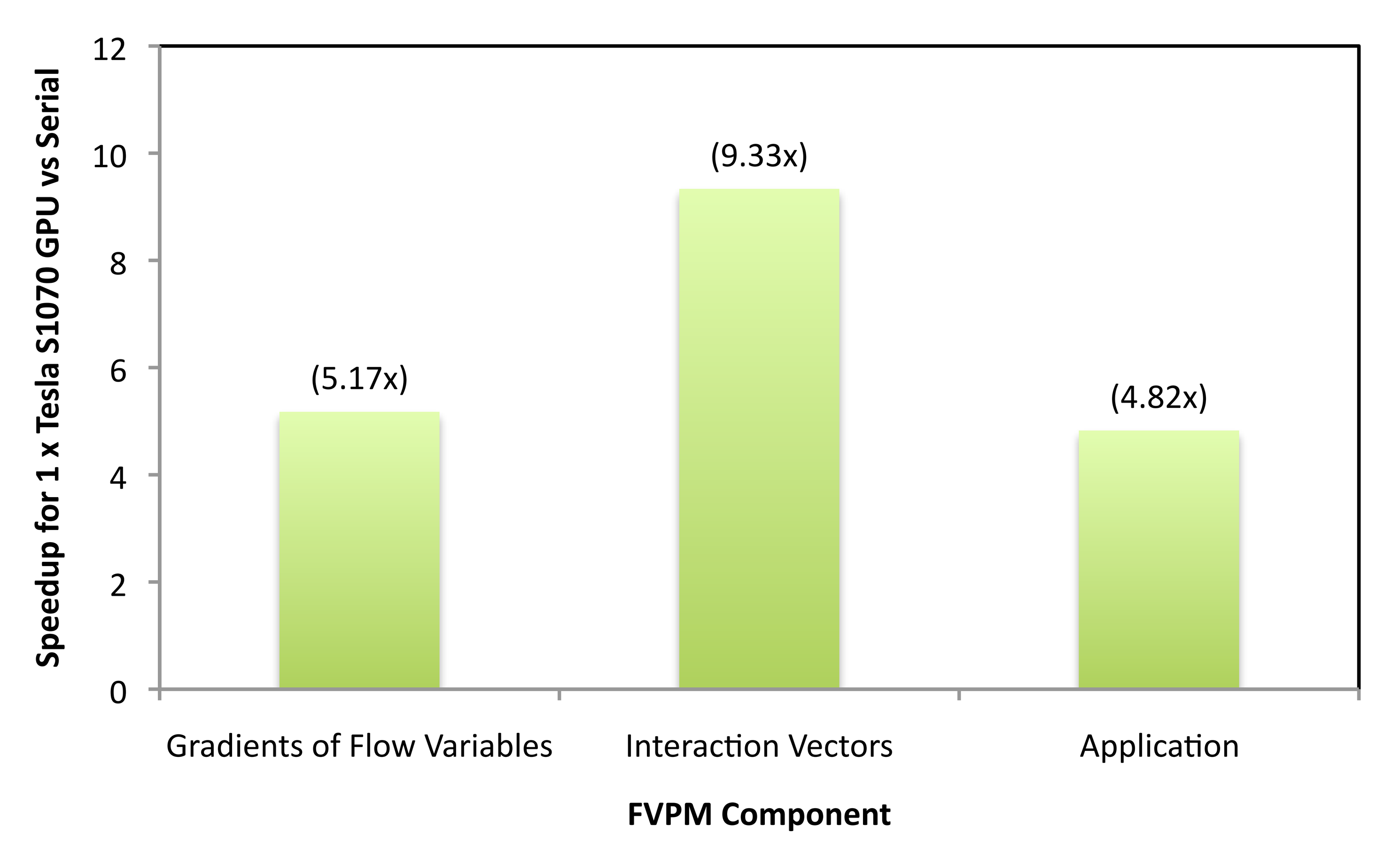
The original serial algorithm consists of an outer loop over all particles which in turn contains two loops over the particle neighbours. Each CUDA thread performs the two neighbour loops for one particle.

```
//Original Serial Looping Structure
for(i=0; i < number_of_particles; i++){
  for(j=0; j < neighbours_of_i; j++){
    //Compute correction matrix (dimensions 2×2)
  }
  //Invert correction matrix
  for(j=0; j < neighbours_of_i; j++){
    //Compute gradient approximation
  }
}
```

Each CUDA thread is assigned two sequential loops over particle neighbours

Results

The performance of the developed CUDA kernels are presented in the below graph. All results are presented for the double precision version of the code with 200,000 FVPM particles.



As additional components of the FVPM code are implemented in CUDA the overall application speedup will be increased. Furthermore, the speedup values reported here are for double precision, and thus are expected to increase significantly for the next-generation Fermi hardware.

References

1. D. Hietel, K. Steiner, and J. Struckmeier, "A Finite Volume Particle Method for Compressible Flows", Mathematical Models and Methods in Applied Science, 10: 1363-1382, 2000
2. R. Nestor, M. Basa, M. Lastiwka and N. Quinlan, "Extension of the finite volume particle method to viscous flow", Journal of Computational Physics, 228:1733-1749, 2009
3. J. Bonet, and T.-S. L. Lok, "Variational and momentum preservation aspects of SPH formulations", Computer Methods in applied mechanics and engineering, 180: 97-115, 1999

ICHEC Irish Centre for High-End Computing
NUI Galway OÉ Gaillimh
Ireland's EU Structural Funds Programmes 2007 - 2013
Co-funded by the Irish Government and the European Union
EUROPEAN REGIONAL DEVELOPMENT FUND
HEA Higher Education Authority An tÚdarás um Ard-Oideachas
Science Foundation Ireland