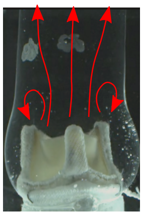
GPU-accelerated Immersed Boundary Method for the efficient simulation of biomedical fluid-structure interactions

Hadi Zolfaghari[1],Barna Becsek[1],Maria Nestola[2],Rolf Krause[2],Dominik Obrist[1]
[1]ARTORG Center for Biomedical Engineering Research, Cardiovascular Engineering, University of Bern, Switzerland
[2]Institute of Computational Science, Faculty of Informatics, University of Lugano, Lugano, Switzerland

UNIVERSITÄT BERN
Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science

Background: the AV-FLOW Project

High-Performance Numerical Simulation of Fluid-Structure Interaction in the aortic heart valve system.



Crucial role in medical device implantation in case of aortic valve disease.

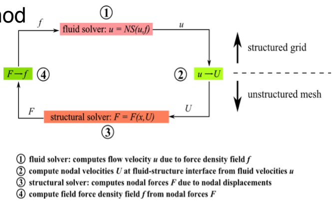
Challenging due to pulsatile turbulent blood flow and elastodynamical behaviour of soft tissue.

Immersed Boundary Method is widely used to avoid expensive mesh-fitting operation of the fluid domain.

Scalable highly accurate incompressible flow solver (IMPACT) simulates the complex blood flow. The elastodynamic deformation of soft tissue is simulated in a dedicated parallel FEM solver (PASSO).

Continuous Immersed Boundary Method

Transfer operations between the fluid grid and the FE mesh are targeted to be accelerated on GPGPUs.

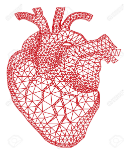


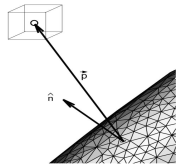
Sharp Interface Immersed Boundary Method

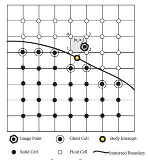
Flow simulation in moving 3D complex geometries on a Eulerian fluid mesh

Sharp-Interface Method is extensively used for biomedical FSI simulations

Ghost Cell Methodology for Immersed Boundary treatment[a]



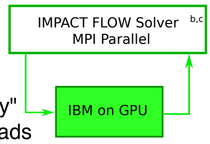




No need for conforming mesh, but the IB treatment is still expensive!

Remedy: Computational Geometry operations can be accelerated on GPUs.

Considerations:
1. CPU-GPU data bandwidth is limited
2. More operation on staggered grid
3. Dynamic allocation is crucial for moving boundaries
4. Parallel reduction is restricted by the "shared memory"
5. Double percision accuracy not achievable on all threads



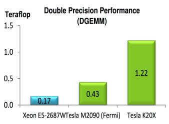
General Purpose Graphical Processing Units(GPGPU)

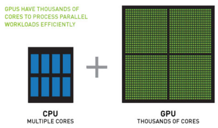
Accelerating floating point operations in shared memory paradigm.

Thousands of parallel threads can be executed on GPU.

Maximum GPU capacity should be utilized to hide the memory-latency.

Tesla K20x GPUs on Cray XC30 machine(CSCS-Piz Daint):
14 multiprocessors
5.6GB GDDR5 memory
max **2048** threads per multiprocessor
max **1024** threads per block





(www.nvidia.com)

Implementation (CUDA FORTRAN by PGI)

CUDA vs. OpenACC

Strong scaling (vs. pragma)

More programming effort

Knowledge of hardware

Tuned choice of thread blocks

Higher speedup!

```
attributes(global) subroutine GPU_Sharp_IB_Method(grid, DOF, IB_geom)
 real, double,texture        ::   coord(*)
 real, double                ::   DOF(:, :, :)
 real,value, shared          ::   IB_geom
 integer                     ::   i,j,k
 i = (blockIdx%x-1)*blockDim%x + threadIdx%x
 j = (blockIdx%y-1)*blockDim%y + threadIdx%y
 k = blockIdx%z
if (k .eq. #DOF_type_ID) then     !on discrete DOF - Staggerred Levels
    //GPU search kernel
endif
end subroutine GPU_Sharp_IB_Method
//declaring device memory
//CUDA memory allocation
call GPU_Sharp_IB_Method<<<Grid,Block>>>(grid,DOF,IB_geom)
//Transfer data back to the CPU
!$cuf kernel do (2) <<<(*,*),(N1p,N2p)>>>
do i=1, N1p
   do j=1, N2p
      do k=1, 3
         if (k .eq. 1) call __staggerred_ib_reductions()
      enddo
   enddo
enddo
```

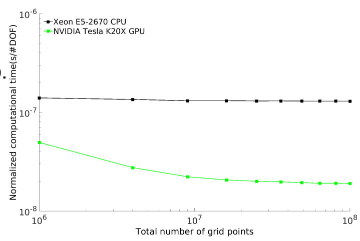
Speedup Measurement

GPU kernels are instructed on a NVIDIA Tesla K20x GPU.

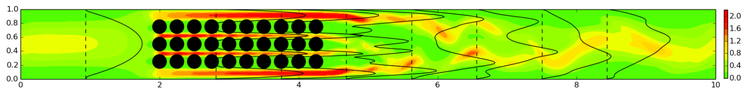
2000x kernel execution speedup.

10x overall speedup for the grid resolutions up to 10^8 grid points.

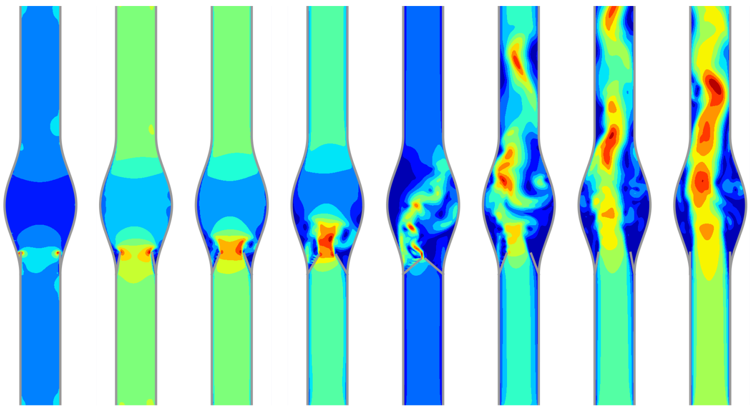
Hiding the host-device bandwith latency by loading the GPU with max threads at a time.



Results



Turbulent incompressible flow through an array of 30 rigid cylinders (Re=10000).



Multi-beat simulation of of turbulent incompressible flow through a mechanical valve (Re=3000).

Extensive IB computations due to:

High-resolution grid for resolving the turbulent flow

Large and discrete solid-fluid interface

Staggerred grid in the flow solver

Acknowledgment

AV-FLOW Project is supported by PASC, the Platform for Advanced Scientific-Computing.

Platform for Advanced Scientific Computing

hadi.zolfaghari@artorg.unibe.ch
www.artorg.unibe.ch
@CVE_ARTORG
CardioVasc Engineers

ARTORG CENTER
BIOMEDICAL ENGINEERING RESEARCH

References
(a) Mittal, R., Dong, H., Bozkurttas M., Najjar, F. M., Vargas, A., von Loebbecke,i A., "A versatile sharp-interface immersed boundary method for incompressible flows with complex boundaries", Journal of-Comptational Physics, 227.10(2008): 4825-4852.
(b) Henniger, R. Obrist, D. and Kleiser, L., "High-order accurate solution of the Navier–Stokes equations on massively parallel computers." Journal of Computational. Physics 299.10 (2010): 3543-3572.
(c) Henniger, R., Obrist, D., and Kleiser, L., "High-order accurate iterative solution of the Navier–Stokes equations for incompressible flows." PAMM 7.1 (2007): 4100009-4100010.