Task-based parallelization of a transport Discontinuous Galerkin solver.

How and why I converted to task-based parallelism

P. Helluy

Inria Tonus, IRMA Strasbourg

SPPEXA Workshop, Garching, 25-27 January 2016

Outlines

Two applications of conservation laws modeling

Implicit DG solver for transport

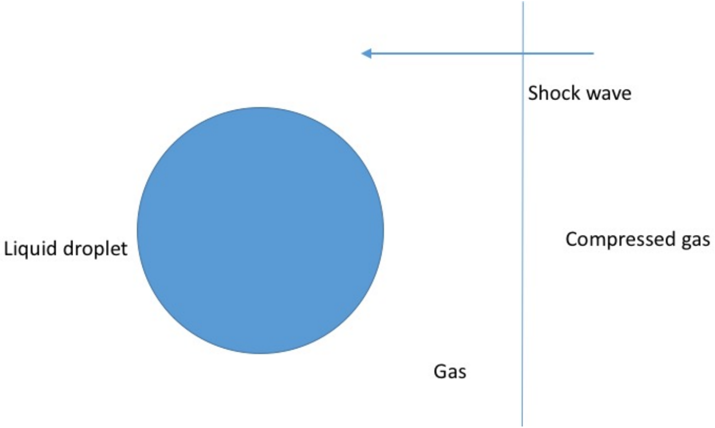
Kinetic conservation laws

Section 1

Two applications of conservation laws modeling

App. I: Shock-droplet interaction

Shock-droplet interaction



Compressible two-fluid model

Vector of conservative variables $W = (\rho, \rho u, \rho v, \rho E, \rho\varphi)^T$, with

- density ρ, velocity U, total energy E.
- color function φ ($\varphi = 0$ in the liquid and $\varphi = 1$ in the gas).

and:

- internal energy $e = E - \frac{u^2+v^2}{2}$.
- pressure law $p = p(\rho, e, \varphi)$.
- flux

$$n \cdot F(w) = (\rho U \cdot n, \rho(U \cdot n)U^T + pn^T, (\rho E + p)U \cdot n, \rho\varphi U \cdot n)^T.$$

System of conservation laws

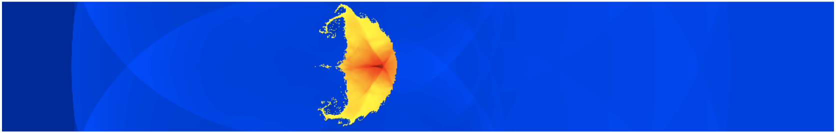
$$\partial_t W + \nabla \cdot F(W) = 0.$$

Cartesian grid solver

- 2D uniform cartesian grid;
- directional Strang splitting;
- Lagrange and remap explicit Finite Volume scheme;
- oscillation free, statistically conservative, Glimm remap [Helluy and Jung, 2014].
- GPU (OpenCL) and multi-GPU (OpenCL+MPI) implementation (5k lines);
- optimized transposition algorithm for better memory bandwidth.

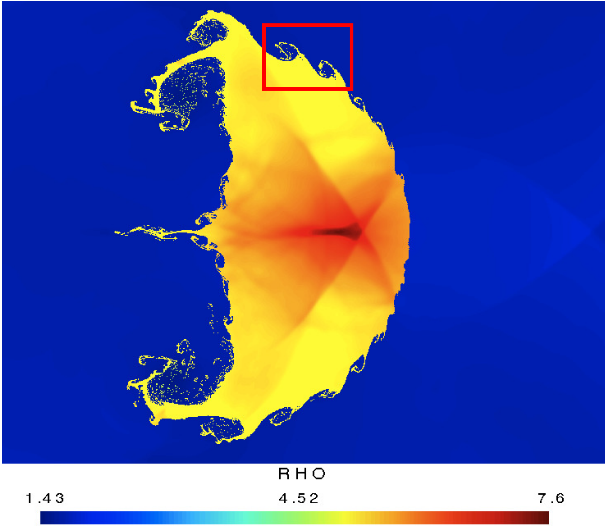
Solution on fine grids

- Very fine mesh OpenCL + MPI simulation, up to 40,000x20,000 grid. 4 billions unknowns per time step
- up to 10xNVIDIA K20 GPUs, $\simeq$30 hours.

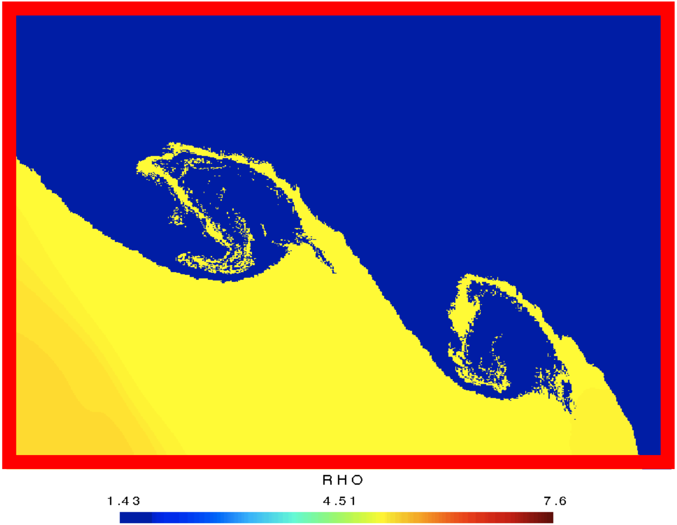


- Same approach also worked for MHD

Solution on fine grids



Solution on fine grids



App. II: Electromagnetic compatibility

Interaction of an EM wave with an aircraft

- Electric field E and magnetic field H
- Maxwell equations

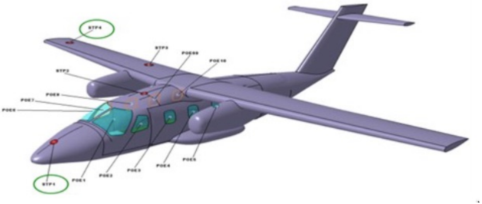
$$\partial_t E - \nabla \times H = 0, \quad \partial_t H + \nabla \times E = 0$$

- Conservative variables $W = (E, H)$, flux $n \cdot F(W) = (-n \times E, n \times H)$
- Again a system of conservation laws

$$\partial_t W + \nabla \cdot F(W) = 0.$$

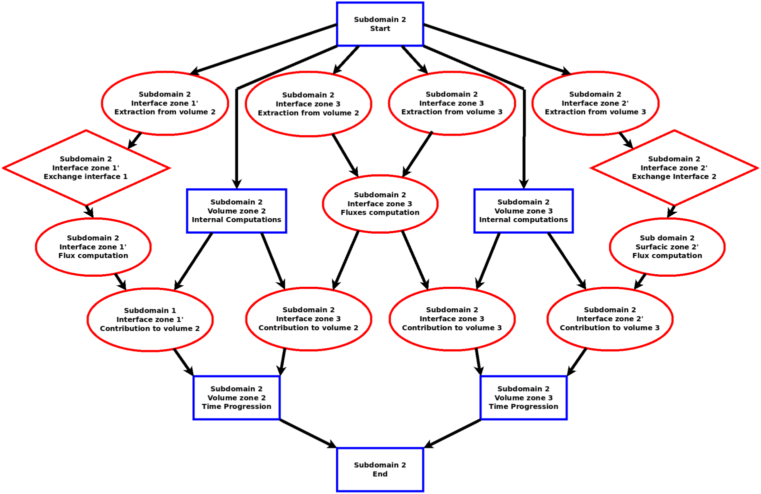
Unstructured grid

More realistic geometries require unstructured grids and more complex parallel implementations.



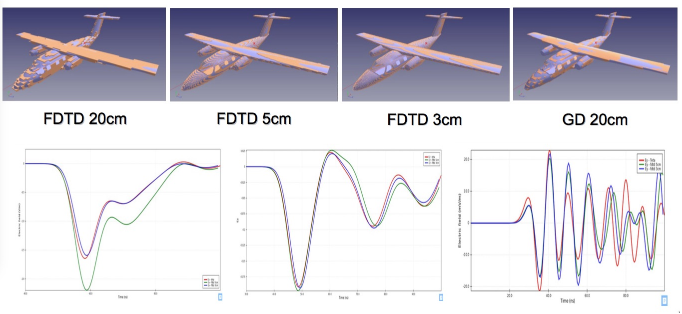
- Discontinuous Galerkin approximation on unstructured grids.
- Multi-GPU (OpenCL + MPI).
- Definition of tasks: source computations, flux computations, communications, etc. with their dependencies.
- Task-based asynchronous implementation (10k lines) [Strub et al., 2015].

Hand-made task graph



Electromagnetic compatibility application

- Electromagnetic wave interaction with an aircraft.
- Aircraft geometry described with up to 3.5M hexaedrons ($\simeq$1 billion unknowns per time step): mesh of the interior and exterior of the aircraft. PML transparent boundary conditions.
- We use 8 GPUs to perform the computation. We achieve 1 TFLOP/s per GPU.



Section 2

Implicit DG solver for transport

Discontinuous Galerkin (DG) interpolation

We consider a coarse mesh made of hexahedral curved macrocells

- Each macrocell is itself split into smaller subcells of size h.
- In each subcell L we consider polynomial basis functions ψ_i^L of degree p.
- G_j^L : Gauss-Lobatto points. Nodal property: $\psi_i^L(G_j^L) = \delta_{ij}$.
- Possible non-conformity in "h" and "p".

On this mesh we solve a simple transport equation with unknown f

$$\partial_t f + v \cdot \nabla f = 0.$$

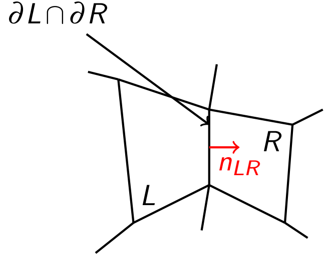
The velocity v is given.

Implicit DG approximation of the transport equation

Implicit DG approximation scheme with upwind flux: $\forall L, \forall i$

$$\int_L \frac{f_L^n - f_L^{n-1}}{\Delta t} \psi_i^L - \int_L v \cdot \nabla \psi_i^L f_L^n + \int_{\partial L} \left(v \cdot n^+ f_L^n + v \cdot n^- f_R^n \right) \psi_i^L = 0.$$

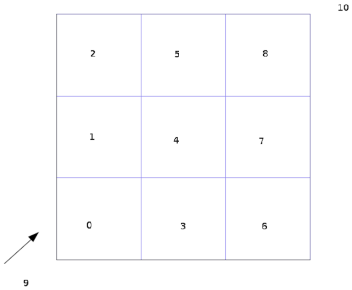
- R denotes the neighbor cells along ∂L.
- $v \cdot n^+ = \max(v \cdot n, 0)$, $v \cdot n^- = \min(v \cdot n, 0)$.
- n_{LR} is the unit normal vector on ∂L oriented from L to R.



Higher order: Crank-Nicolson, diagonally implicit Runge-Kutta, etc.

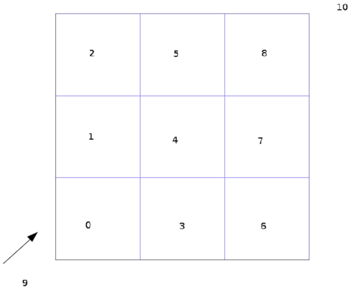
Upwind numbering

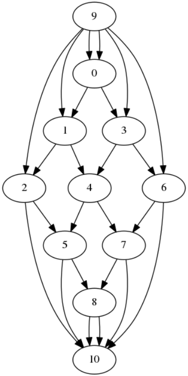
- L is *upwind* with respect to R if $v \cdot n_{LR} > 0$ on $\partial L \cap \partial R$.
- In a macrocell L, the solution depends only on the values of f in the upwind macrocells.
- No assembly and factorization of the global system.



Dependency graph

For a given velocity v we can build a dependency graph. Vertices are associated to macrocells and edges to macrocells interfaces or boundaries. We consider two fictitious additional vertices: the "upwind" vertex and the "downwind" vertex.





Algorithm

[Duff and Reid, 1978, Johnson et al., 1984, Wang and Xu, 1999, Natvig and Lie, 2008]

- Topological ordering of the dependency graph.
- First time step: Assembly and LU decomposition of the local macrocell matrices.
- For each macrocell (in topological order):
  - Compute volume terms.
  - Compute upwind fluxes.
  - Solve the local linear system.
  - Extract the results to the downwind cells.

Parallel implementation ?

StarPU

- StarPU is a library developed at Inria Bordeaux [Augonnet et al., 2012]: `http://starpu.gforge.inria.fr`
- Task-based parallelism.
- Task description: codelets, inputs (R), outputs (W or RW).
- The user submits tasks in a correct sequential order.
- StarPU schedules the tasks in parallel (if possible) on available cores and accelerators.
- MPI still needed for large scale computations.

Preliminary results

We compare a global direct solver to the upwind StarPU solver with several meshes.
Weak scaling. "dmda" scheduler. AMD Opteron 16 cores, 2.8 Ghz.
Timing in seconds for 200 iterations.

nb cores	0	1	2	4	8	16
$10 \times 10 \times 8 \times 8$ direct	30	144	-	-	-	-
$10 \times 10 \times 8 \times 8$ upwind	-	32	19	12	7	6
$20 \times 20 \times 4 \times 4$ upwind	-	41	26	17	12	17
$20 \times 20 \times 8 \times 8$ upwind	-	120	72	40	28	20

Section 3

Kinetic conservation laws

Framework

- Distribution function: $f(x, v, t)$, $x \in \mathbb{R}^d$, $v \in \mathbb{R}^d$, $t \in [0, T]$.
- Microscopic "collision vector" $K(v) \in \mathbb{R}^m$. Macroscopic conserved data

$$w(x,t) = \int_v f(x,v,t)K(v)dv.$$

- Entropy $s(f)$ and associated Maxwellian $M_w(v)$:

$$\int_v M_w K = w, \quad \int_v s(M_w) = \max_{\int_v fK = w} \left\{ \int_v s(f) \right\}.$$

- Transport equation ($a = a(x,t)$ is the acceleration) with relaxation:

$$\partial_t f + v \cdot \nabla_x f + a \cdot \nabla_v f = \eta \left(M_w - f \right).$$

Kinetic schemes

When the relaxation parameter η is big, the Vlasov equation provides an approximation of the hyperbolic conservative system

$$\partial_t w + \nabla \cdot F(w) + S(w) = 0,$$

with

$$F^i(w) = \int_v v^i M_w(v) K(v) dv.$$

$$S(w) = a \cdot \int_v \nabla_v M_w(v) K(v) = -a \cdot \int_v M_w(v) \nabla_v K(v).$$

Main idea: numerical solvers for the linear scalar transport equation lead to natural solvers for the non-linear hyberbolic system [Deshpande, 1986]. Micro or macro approach.

Applications

The following models enter this framework:

- Compressible flows [Perthame, 1990]
- lattice Boltzmann schemes for low Mach flows [Qian et al., 1992]
- lattice Boltzmann schemes for MHD [Dellar, 2002]
- and even Maxwell equations !

The underlying kinetic model has not necessarily a physical meaning.

Conclusion & future works

- Migration of a transport DG solver to StarPU. Comfortable task-based parallelism.
- SCHNAPS (“Solveur Conservatif Non-linéaire Appliqué aux PlaSmas”) `http://schnaps.gforge.inria.fr` (40k lines)
- Future works within EXAMAG:
  - StarPU codelets for GPU (OpenCL or CUDA).
  - MPI + StarPU.
  - Kinetic schemes, Vlasov, MHD.

Bibliography I

[Augonnet et al., 2012] Augonnet, C., Aumage, O., Furmento, N., Namyst, R., and Thibault, S. (2012).
StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators.
In Jesper Larsson Träff, S. B. and Dongarra, J., editors, *EuroMPI 2012*, volume 7490 of *LNCS*. Springer.
Poster Session.

[Dellar, 2002] Dellar, P. J. (2002).
Lattice kinetic schemes for magnetohydrodynamics.
Journal of Computational Physics, 179(1):95–126.

[Deshpande, 1986] Deshpande, S. (1986).
Kinetic theory based new upwind methods for inviscid compressible flows.
In *24th AIAA Aerospace Sciences Meeting*, volume 1.

[Duff and Reid, 1978] Duff, I. S. and Reid, J. K. (1978).
An implementation of tarjan's algorithm for the block triangularization of a matrix.
ACM Transactions on Mathematical Software (TOMS), 4(2):137–147.

[Helluy and Jung, 2014] Helluy, P. and Jung, J. (2014).
Interpolated pressure laws in two-fluid simulations and hyperbolicity.
In *Finite volumes for complex applications. VII. Methods and theoretical aspects*, volume 77 of *Springer Proc. Math. Stat.*, pages 37–53. Springer, Cham.

[Johnson et al., 1984] Johnson, C., Nävert, U., and Pitkäranta, J. (1984).
Finite element methods for linear hyperbolic problems.
Computer methods in applied mechanics and engineering, 45(1):285–312.

[Natvig and Lie, 2008] Natvig, J. R. and Lie, K.-A. (2008).
Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements.
Journal of Computational Physics, 227(24):10108–10124.

Bibliography II

[Perthame, 1990] Perthame, B. (1990).
Boltzmann type schemes for gas dynamics and the entropy property.
SIAM Journal on Numerical Analysis, 27(6):1405–1421.

[Qian et al., 1992] Qian, Y., d'Humières, D., and Lallemand, P. (1992).
Lattice bgk models for navier-stokes equation.
EPL (Europhysics Letters), 17(6):479.

[Strub et al., 2015] Strub, T., Helluy, P., Massaro, M., and Roberts, M. (2015).
Asynchronous OpenCL/MPI numerical simulations of conservation laws.
working paper or preprint.

[Wang and Xu, 1999] Wang, F. and Xu, J. (1999).
A crosswind block iterative method for convection-dominated problems.
SIAM Journal on Scientific Computing, 21(2):620–645.