SeLaLib (Semi-Lagragian Library)

TONUS Project Team

Pierre Navaro (IRMA Strasbourg)

C2S@Exa Inria Project Lab (IPL)
Technical meeting Sparse linear system solvers

23 septembre 2013

Outline

- History
- Context :
  - plasma physics and the mathematical model
  - the model and its requirements
  - research topics
- Library modules overview
- Abstraction examples
  - deformed meshes
  - remap parallel module
- Ongoing work
- Software development remarks
- Perspective and needs.

History

- 2001 Eric Sonnendrucker at IRMA
- 2004 CALVI project begins
- 2007 ANR EGYPT (Etude GYrocinétique des Plasmas Turbulents)
- 2009 AE FUSION
- 2010 ANR GYPSI (GYrokinetic high Performance Simulation for ITER)
- 2010 Vlasy (python platform)
- 2010 ADT SeLaLib
- 2013 TONUS (Tokamaks and Numerical Simulations) lead by Philippe Helluy

Collaborative projects with Institut de Recherche sur la Fusion Magnétique (CEA, Cadarache)

The Physics Model and its Requirements

- The fundamental equation of plasma physics is the Vlasov equation :

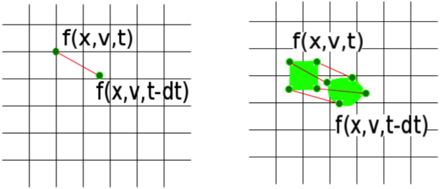
$$\frac{\partial f_s(\vec{x},\vec{v},t)}{\partial t} + \vec{v}\cdot\nabla_{\vec{x}} f_s + \frac{q_s}{m_s}(\vec{E}+\vec{v}\times\vec{B})\cdot\nabla_{\vec{v}} f_s = 0.$$

- Its solution is sought in a variety of conditions :
  - from 2D to 6D problems,
  - electrostatic or electromagnetic approximations, gyrokinetic approximations... (i.e : more coupled equations),
  - flexibility of the choice of solution methods,
  - arbitrary shapes of the solution domains...
- Selalib is meant to provide a collection of software modules to aid in the development of simulations configurable in any combination of the above...

Context : Research Topics

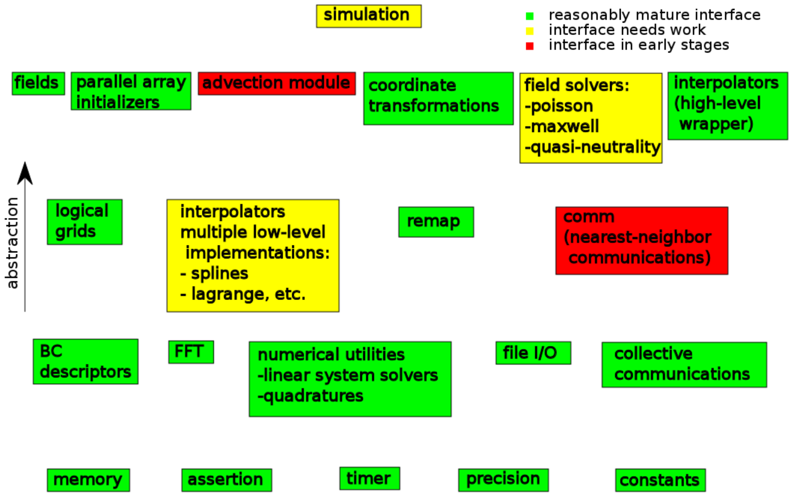
- The main types of equation that result from splitting the Vlasov equation are a) constant coefficients, b) non-conservative and, c) conservative form :

$$\begin{aligned}
\partial_t f(x,v,t) + a\partial_x f(x,v,t) &= 0 \\
\partial_t f(x,v,t) + a(x,t)\partial_x f(x,v,t) &= 0 \\
\partial_t f(x,v,t) + \partial_x (a(x,t) f(x,v,t)) &= 0
\end{aligned}$$



- Ongoing work is carried out to develop such methods and interface them to the library.

Library Modules Overview (not comprehensive)



Low-Level Modules

- At the lowest level we have services which are general, fully reusable and do not depend on any other module :
  - memory manager : basic allocators/deallocators,
  - assertions : simple, defensive programming
  - working precision : basic numeric types, C bindings.
  - constants : physical and math,
  - timer : timing utility
  - I/O routines (HDF5 or binary format)
  - utilities : for miscellaneous low-level functions (is_power_of_two(), int2string(), etc.)

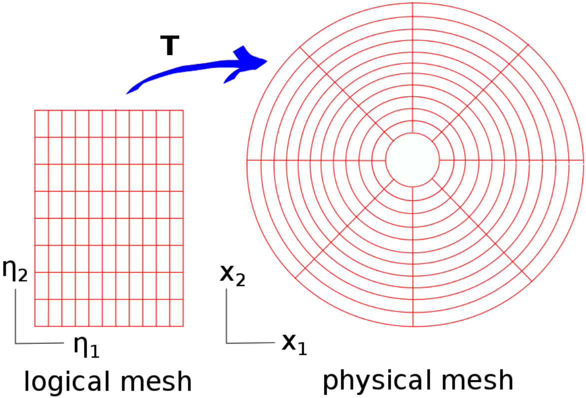
Mid-Level : mathematical and parallel services

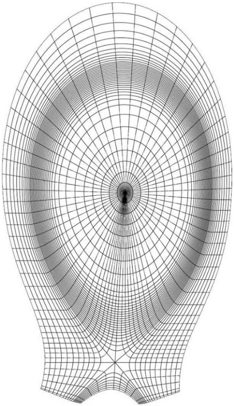
- tridiagonal system solver,
- numerical integration,
- splines : cubic 1D, 2D, nonuniform, in progress : quintic splines,
- FFT (generic interface with multiple implementations),
- nonlinear solvers (Newton-Raphson),
- ODE solvers : RK for specific ODEs,
- collective module : the entry point for MPI,
- remap module : for global redistribution of data,
- comm module : for nearest neighbor parallelization schemes,
- field solvers (for array data)

Top-Level Modules

- Physics-related capabilities :
  - interpolators : generic wrapper for cubic splines, etc.,
  - coordinate transformations,
  - fields (scalar and vector) : logical mesh + transformation + data,
  - integrated simulations
  - specialized solvers (quasi-neutrality equation)
- We have been using the features of the Fortran 2003 standard to aid in this development, hence paid a portability price.
- Eventually, some modules will be downgraded or will have multiple implementations in specific cases, for example having both F95 and F2003 implementations.

Abstraction Example (I)





- Arbitrary physical domains are modeled by structured meshes acted upon by a coordinate transformation.
- Reasonably complex configurations can be modeled by patching multiple structured meshes.

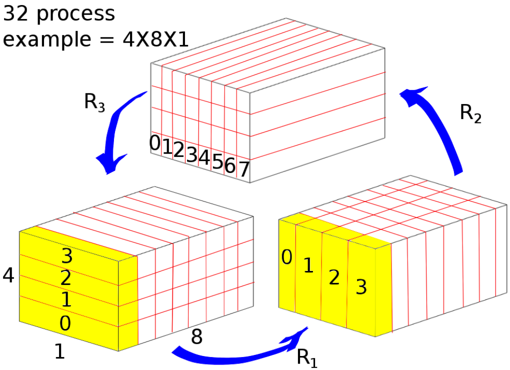
Abstraction Example (II)

sll_logical_mesh_2D	sll_coordinate_transformation_2D
num_cells_eta1	x1(η_1,η_2)
num_cells_eta2	x2(η_1,η_2)
eta1_min	jacobian(η_1,η_2)
eta1_max	jacobian_matrix(η_1,η_2)
delta_eta1	inverse_jacobian_matrix(η_1,η_2)
eta2_min	x1_at_node(i,j)
eta2_max	x2_at_node(i,j)
delta_eta2	jacobian_at_node(i,j)
	x1_at_cell(i,j)
	x2_at_cell(i,j)
	jacobian_at_cell(i,j)
	write_to_file( )
	delete( )

- The coordinate transformation initialization depends on its being of the analytical or discrete type.

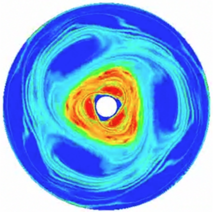
Abstraction Example : Remap Parallel Module

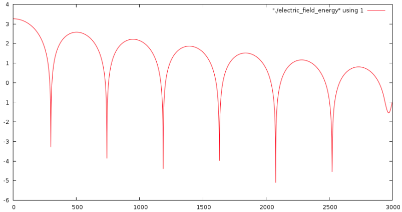
```
type(layout_3d) :: lx,ly; type(remap_3d)  :: r1
call initialize_layout(np1, np2, np3, 1, 4, 8, lx)
call initialize_layout(np1, np2, np3, 4, 1, 8, ly)
r1 => new_remap_plan( lx, ly, array_x )
call apply_remap_plan( r1, array_x, array_y )
```



Ongoing Work

- In development : multiple instances of our simulation class :
  - 4D parallel, (x, y, v_x, v_y) : tested
  - 4D drift-kinetic $(x, y, z, v_{||})$: polar coordinates
  - 4D drift-kinetic $(x, y, z, v_{||})$, with finite-volume advection scheme
  - 4D euler scheme with finite elements.
- alternative parallel module, for nearest-neighbor communications
- ongoing research in the development of advection schemes, and test cases for simulations.





Software Development Remarks

- IS : **Edwin Chacon-Golcher** (Software designer)
- Postdoc : Aurore Back, Adnane Hamiaz
- IJD : **Raphael Blanchard**, Samuel Desantis, Aliou Diouf
- IR : Pierre Navaro
- Mathematicians : Philippe Helluy, Michel Mehrenberger, Eric Sonnendrucker.
- Others mathematicians developing algorithms both inside and outside the library as users.

Software engineering

- git + CMake + CTest + CDash
- Fortran 2003 + MPI + HDF5 + FFTW
- INRIA provides GForge + Continuous Integration portal + CMake quality dashboard


Login All Dashboards — Monday, June 17 2013 10:53:01 CEST

Selalib

Dashboard Calendar Previous Current Next Project

This is the new CDash version

No file changed as of **Monday, June 10 2013 - 21:00 CEST** — Show Filters Advanced View Auto-refresh Help

Nightly

Site	Build Name	Update Files	Configure Error	Configure Warn	Build Error	Build Warn	Test Not Run	Test Fail	Test Pass	Build Time
irma-suse.u-strasbg.fr	gfortran-4.7.3-Linux-x86_64	0	0	0	0	0	0	7	54	Jun 11, 2013 - 00:00 CEST
irma-hpc	ifort-13.1.2-Linux-x86_64	0	0	0	0	0	0	3	58	Jun 11, 2013 - 00:00 CEST
PIPOL	gfortran-4.6.3-amd64-linux-ubuntu-precise	0	0	0	0	0	0	2	59	Jun 11, 2013 - 08:54 CEST
PIPOL	gfortran-4.7.2-amd64-linux-ubuntu-quantal	0	0	0	0	0	0	2	59	Jun 11, 2013 - 09:02 CEST
PIPOL	gfortran-4.7.2-amd64_2010-linux-fedora-core17	0	0	0	0	0	0	2	59	Jun 11, 2013 - 08:54 CEST
PIPOL	gfortran-4.7.3-amd64-linux-debian-testing	0	0	0	0	0	0	2	59	Jun 11, 2013 - 08:53 CEST
PIPOL	gfortran-4.7.3-amd64_2010-linux-debian-testing	0	0	0	0	0	0	2	59	Jun 11, 2013 - 08:41 CEST
irma-hpc2	gfortran-4.7.3-Linux-x86_64	0	0	0	0	0	0	2	59	Jun 11, 2013 - 00:00 CEST

Experimental

Site	Build Name	Update Files	Configure Error	Configure Warn	Build Error	Build Warn	Test Not Run	Test Fail	Test Pass	Build Time
irma-suse.u-strasbg.fr	gfortran-4.7.3-Linux-x86_64		0	0	0	0	0_{-19}	2^{+1}	59^{+18}	Jun 11, 2013 - 14:25 CEST
irma-suse.u-strasbg.fr	gfortran-4.7.3-Linux-x86_64		0	0	0	0	19	1	41	Jun 11, 2013 - 14:13 CEST
mp-navaro.local	gfortran-4.8.1-Darwin-i386		0	0	0	0	0	1	53	Jun 10, 2013 - 21:25 CEST

CDash 2.0.2 © Kitware | Report problems | 0.084s

Perspectives

- First release planned in January 2014.
- Future planned additions include :
  - domain-decomposition,
  - add support modules : logger, checkpointer, etc.
  - introduce hardware acceleration techniques.
- Ideally, Selalib would become a trusted collection of reusable tools for building plasma physics simulations.
- Selalib should provide a common framework where geographically distributed researchers would be able to find :
  - tools for their work,
  - opportunities to contribute/collaborate with others while using defined software processes instead of *ad hoc* developments.

Linear solver : What we have

- Fast band matrix solver for interpolation routines (Lapack).
- FFT Poisson solvers (home made or Fishpack)
- General solver for curvilinear meshes (Multigrid or conjuguate gradient method)
- Variational Integrators method needs non linear solvers (Newton + preconditioned conjugate gradient method)
- Exponential integrators for spectral element, discontinuous Galerkin method. (Suitesparse library)

Linear solver : What we need and consider

- Prerequisites
  - Preconditioning tool
  - Parallel solvers
  - Fortran interface
  - Non commercial
  - No exotic dependencies (ideally)
- We consider
  - Pastix + Scotch + Murge
  - PETSc ?