Polyphemus Training Day

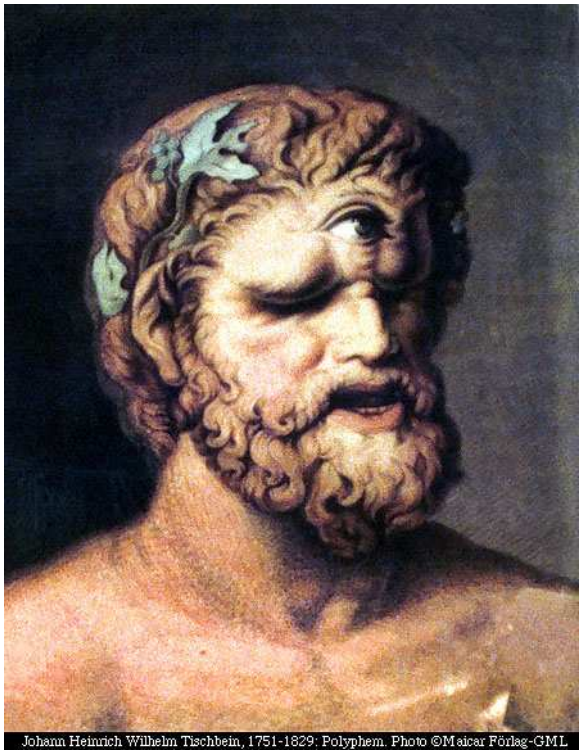
Introduction to Polyphemus (translated from French)

Vivien Mallet, for the development team

27 March 2007

Outline

Polyphemus Images



Greek Mythology

- Polyphemus, cyclops in Odyssey

Why this name ?

- « Poly » : multiple
- « phemus » : speech

Multiple Goals

Multiple Models

- Scales : from local scale to continental scale
- Formulations : Gaussian, Eulerian, . . .

Multiple Pollutants

- Passive, radionuclides
- Photochemistry
- Aerosols
- Persistent organic pollutants, heavy metals, . . .

Multiple Inputs

- From meteorological models
- Ground data

Multiple Methods

Data Assimilation

- Sequential
- Variational
- Inverse modeling (parameter estimation)

Ensemble Forecast

- Multimodels
- Monte Carlo
- Models combinations (« superensembles », . . .)

Models Coupling

- Feedbacks
- Impact

Constraints

Perennial Code

- System maintenance
- Scalable, integration of new developments

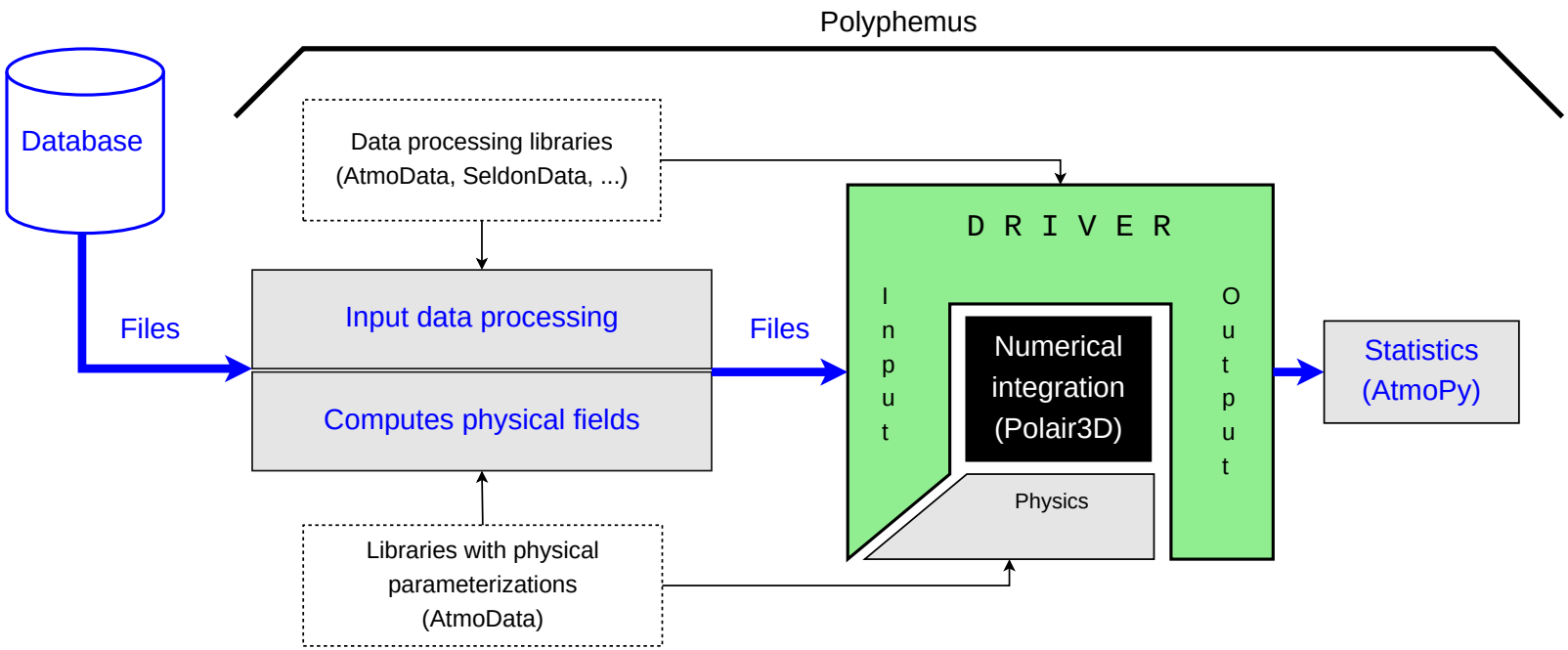
Open

- Availability, distribution
- Development or contributions from other teams

Field Context

- From research to operational use

Overall structure



Programming Choices

Main language : C++

- Efficient for scientific computing
- Advanced object design
  - Inheritance, genericity
  - Management of complex objects
  - Exceptions
- Widely used and perennial

Example : Data Manipulation

```
LinearInterpolationRegular(SoilWater, SoilWater_out);
SoilWater_out.Threshold(0., 1.);
```

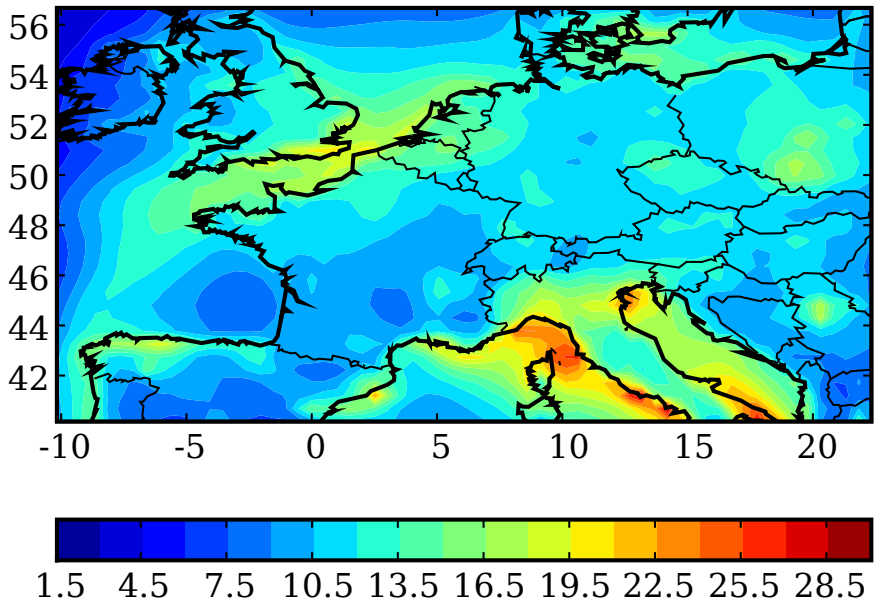
Example : Model Manipulation

```
Model.Forward();
OutputSaver.Save(Model);
```

Programming Choices

Complementary Language : Python

- Dynamic, interactive
- Visualization
- Scripts
- Increasingly used in scientific computing



Programming Choices

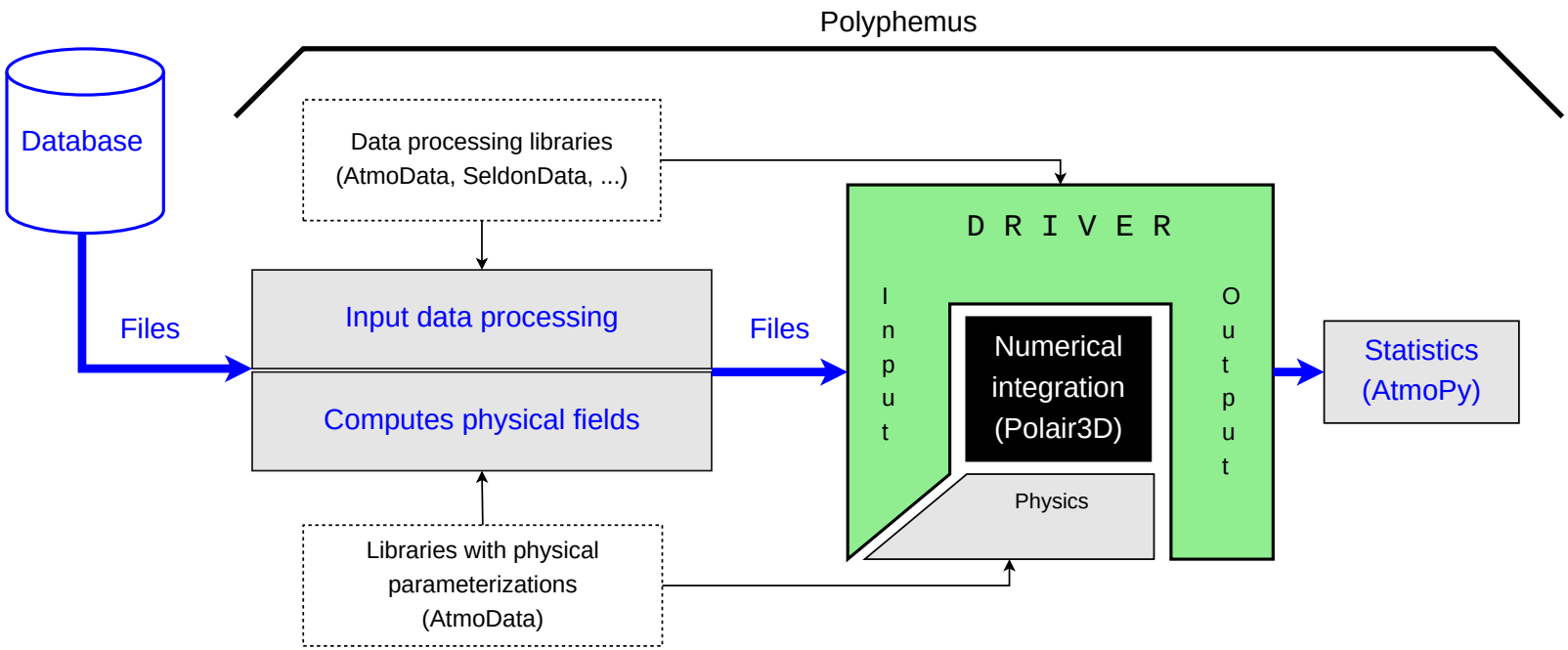
Historical Language : Fortran 77

- Automatic differentiation
- Continuity
- Calls from C++

Policy

- To avoid dealing with too many languages
- To use primarily languages with strong potential and productivity

Overall Structure



Libraries

AtmoData, SeldonData, AtmoPy, Talos

Preprocessing, Data Management

- SeldonData (C++)
  - Interpolations, input/output operations, . . .
- AtmoData (C++)
  - Extension of SeldonData to atmospheric sciences

Physics

- AtmoData (C++, Fortran 77)
  - Meteorology, emissions, . . .

Visualization, Postprocessing, Statistics

- AtmoPy (Python)

Miscellaneous

- Talos (C++) : configuration files

Models

Gaussian Models

- Stationary Gaussian model and puff Gaussian model
- Gas and aerosols
- Several parameterizations to compute the dispersion

Eulerian Models

- Castor (clone of the gas version of Chimere)
  - Passive and chemical versions
- Polair3D
  - Passive, chemical, aerosol and adjoint (for gas) versions
- Modules for transport, chemistry and aerosols

Modules

Transport

- Advection (DST3, PPM), diffusion (ROS2)

Chemistry

- RADM, RACM, Melchior
- Radioactive or biological decay

Aerosols

- SIREAM (semi-Lagrangian)

Output Savers

- Whole domain, list of points
- Boundary conditions (for nesting)
- Deposition fluxes, . . .

Preprocessing

Meteorological Fields

- ECMWF, MM5
- Vertical diffusion : Louis and Troen&Mahrt

Ground Data

- Land use cover : USGS, GLCF
- Emissions from EMEP
- Biogenic emissions (Simpson et al., 1999)
- Deposition : Emberson, Wesely and Zhang

Boundary Conditions

- Mozart 2, Inca
- Gocart

Gaussian Models

Drivers

Forward Simulation (`BaseDriver`)

- A model is a C++ object

```
/*** Initializations ***/

Model.Init();
OutputSaver.Init(Model);

/*** Time loop ***/

for (int i = 0; i < Model.GetNt(); i++)
  {
    Model.InitStep();
    OutputSaver.InitStep(Model);

    Model.Forward();
    OutputSaver.Save(Model);
  }
```

Drivers

Optimal Interpolation

```
for (int i = 0; i < Model.GetNt(); i++)
  {
        Model.InitStep();
        OutputSaver.InitStep(Model);

        Model.Forward();

        OutputSaver.SetGroup("forecast");
        OutputSaver.Save(Model);

        // Retrieves observations.
        ObsManager.SetDate(Model.GetCurrentDate());

        if (ObsManager.IsAvailable())
          {
            Model.GetState(state_vector);
            Analyze(state_vector);
            Model.SetState(state_vector);
          }
  }
```

Drivers

Forward Simulation

- `BaseDriver`, `StationaryDriver`
- `PlumeDriver`, `PuffDriver`

Data Assimilation

- Optimal interpolation
- Kalman filters : ensemble version et reduced-rank version
- 4D-Var (and adjoint validation)

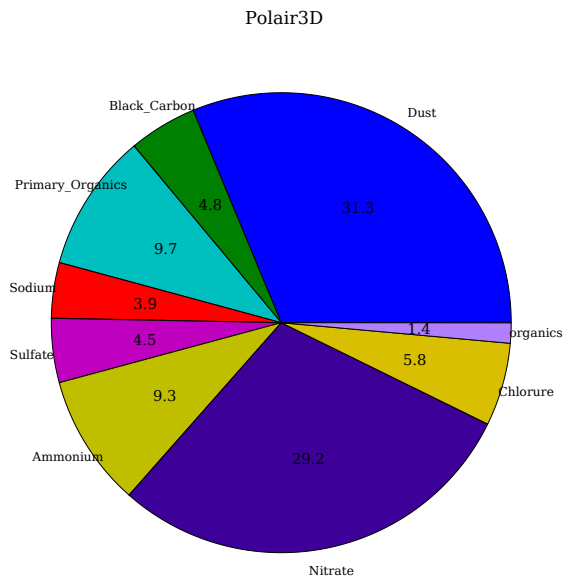
Under development

- Plume-in-grid model
- Monte Carlo
- Models coupling : soil model / atmospheric model

Postprocessing

Programs

- Comparisons to observations (error statistics, visualization)
- Water diagnosis in a plume



AtmoPy Library

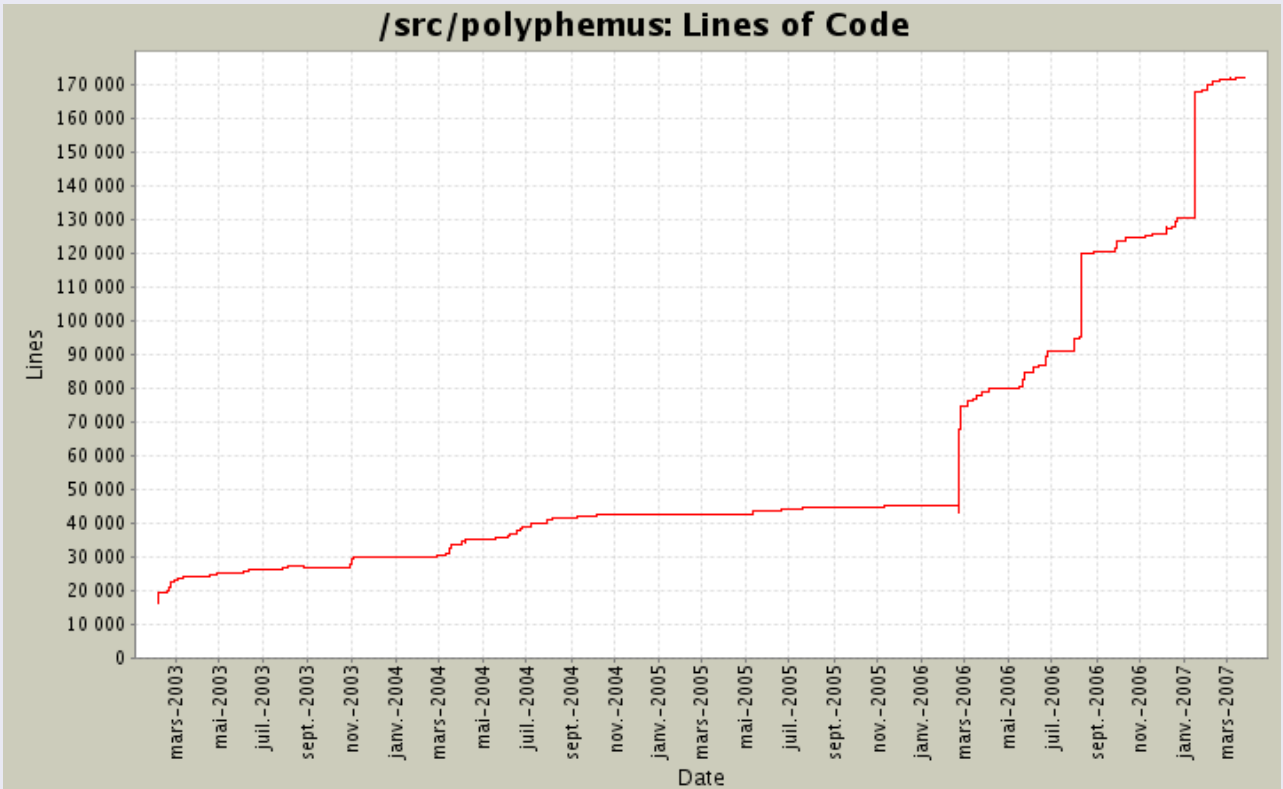
- Graphical visualization
- Observations management
- Statistical measures
- Ensembles management (models combinations)

Resources

Code

- 50 000 lines of hand-written code (SLOCcount)
- 50 000 lines automatically generated

Subversion Repository (without SIREAM)



Resources

Current Developers

1	Meryem Ahmed de Biasi (INRIA)	diffusion
2	Édouard Debry (ENPC)	aerosols
3	Karine Kata-Sartelet (ENPC)	aerosols
4	Irène Korsakissok (ENPC)	local
5	Vivien Mallet (ENPC)	ensemble
6	Denis Quélo (IRSN)	passive
7	Yelva Roustan (ENPC)	impact
8	Bruno Sportisse (ENPC)	aerosols
9	Marilyne Tombette (ENPC)	aerosols
10	Lin Wu (INRIA)	assimilation

Resources

Documentations for users

- User's guide (140 pages)
- Scientific documentation for AtmoData
- Reference documentation for AtmoPy

Documentations for developers

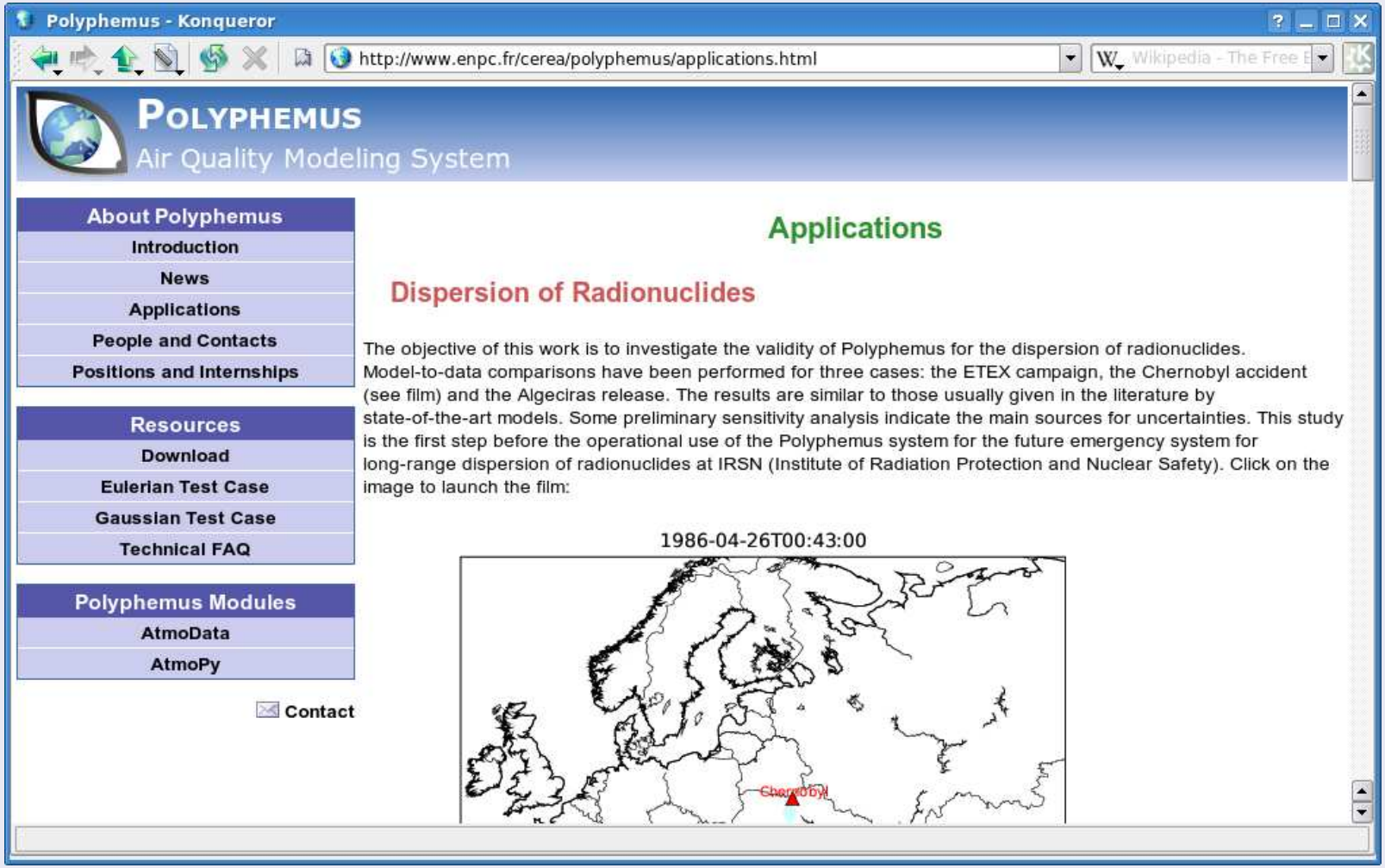
- Guide and reference documentation for SeldonData
- Reference documentation for AtmoData
- Reference documentation for Talos

Examples

- Test cases (Eulerian and Gaussian)
- Practical sessions (primarily for courses at ENSTA and ENPC)

Resources

http://www.enpc.fr/cerea/polyphemus/



Next...

Models (mostly implemented, but not yet included in Polyphemus)

- Secondary organic aerosols
- Modal aerosol module (MAM)
- Heavy metals, mercury
- Passive hemisphere model

Next Steps

- Parallelization
- Lagrangian particle model
- Drivers for assimilation and coupling