Assimilation de données dans Polyphemus

Action ADOQA

Lin Wu et Vivien Mallet

Lin.Wu@inria.fr

Projet CLIME (INRIA / ENPC)

Introduction

System structure of Polyphemus

Data assimilation in Polyphemus

Preliminary results and future developments

Présentation de Polyphemus

Introduction

- Système pour la modélisation de la qualité de l'air
- Multiples échelles (locale – $\sim 100\,\mathrm{m}$ –, régionale, continentale)
- Repose sur
  - la gestion de données,
  - des paramétrisations physiques,
  - un cœur numérique pour une équation d'advection-diffusion-réaction :

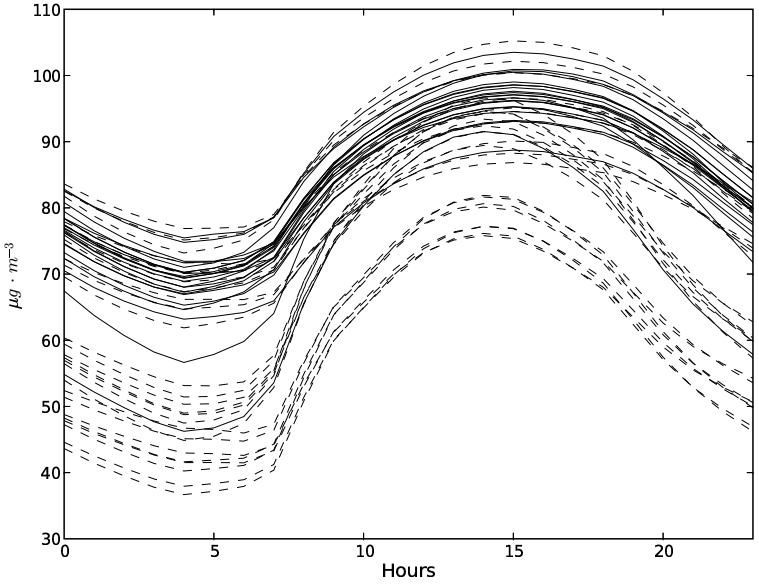
$$\frac{\partial c_i}{\partial t} = -\mathrm{div}(V c_i) + \mathrm{div}(K \nabla c_i) + \chi_i(c) + S_i - L_i$$

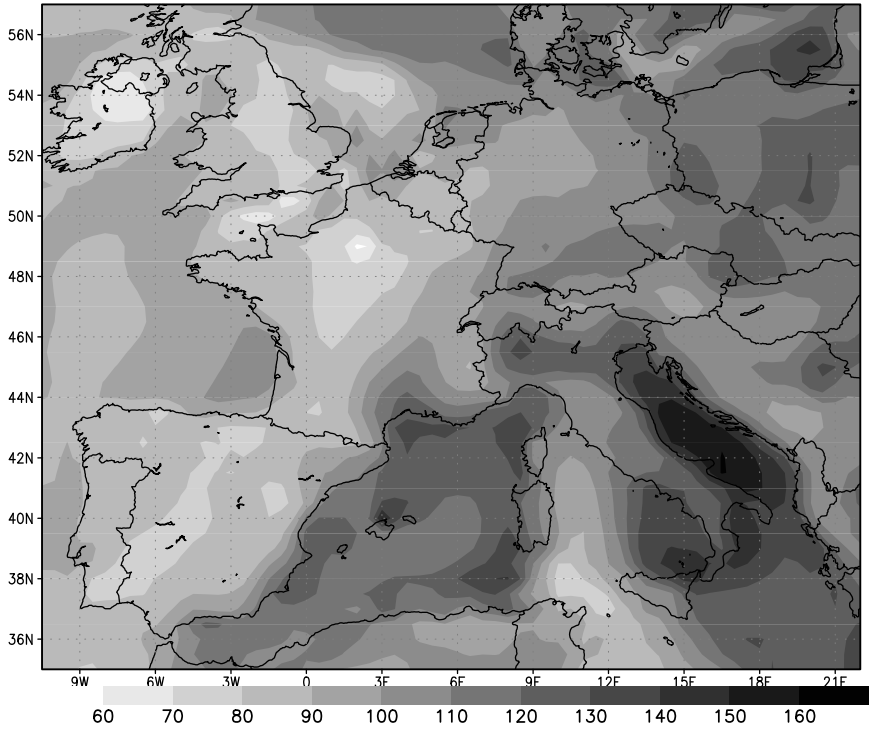
$$K \nabla c_i = E_i - v_i c_i$$

- 10^5 à 10^7 variables suivies en temps

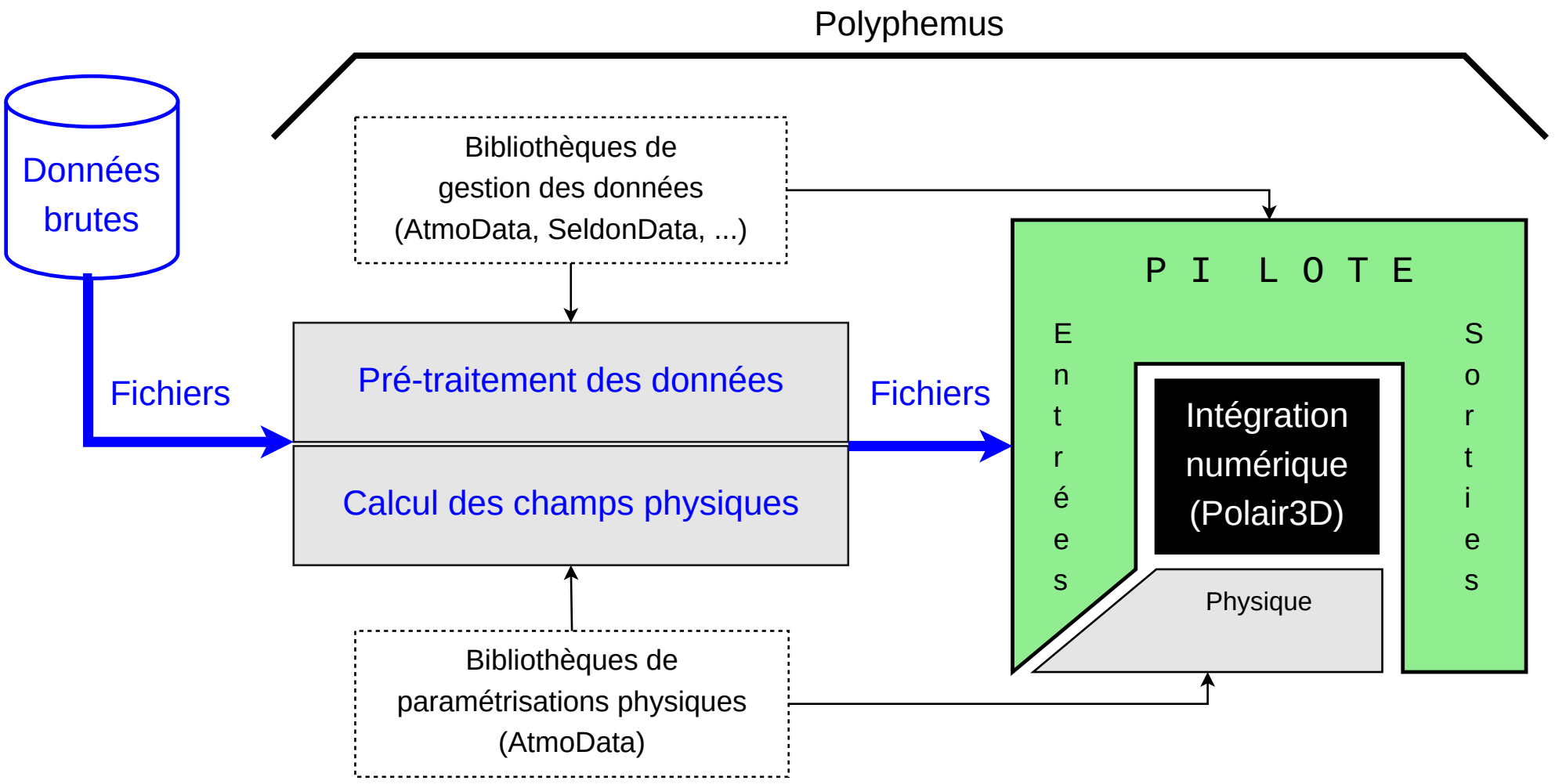
Présentation de Polyphemus

Exemple : ozone à l'échelle continentale



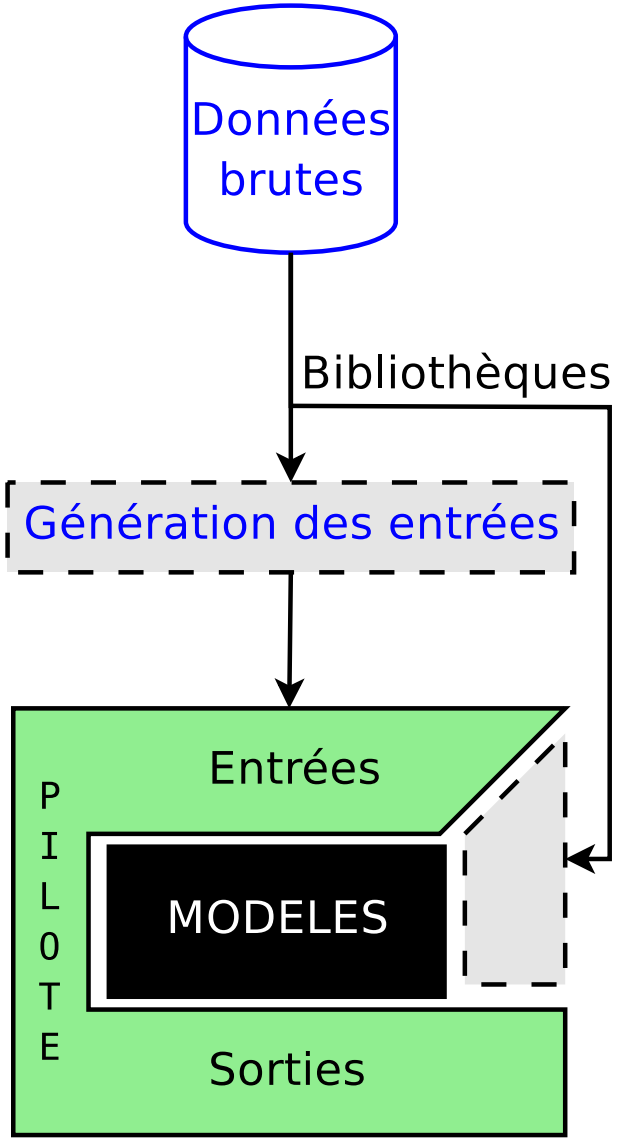


Architecture de Polyphemus



Architecture de Polphemus

Assimilation de données



- Modèles sous forme d'objets C++
- Pilotes
  - Un pilote par méthode d'assimilation
  - Appels à l'interface des modèles (méthodes)

Architecture de Polphemus

Objets

- Modèle : Forward, GetState, ...
- Observations : MltObsOperator, TLMMltObsOperator, ...

Exemple : pilote direct

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

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

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

Architecture de Polphemus

Exemple : Interpolation optimale

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

    this->Model.Forward();

    // Retrieves observations.
    this->ObsManager.SetDate(this->Model.GetCurrentDate());

    if (this->ObsManager.IsAvailable())
      {
        this->Model.GetState(state_vector);
        Analyze(state_vector);
        this->Model.SetState(state_vector);
      }

    this->OutputSaver.Save(this->Model);
  }
```

Architecture de Polphemus

Exemple : Interpolation optimale

```
// Computes HBH'.
for (c = 0; c < this->Nobs; c++)
  {
    column = 0.;
    // Computes the column #c of BH'.
    for (j = 0; j < this->Nstate; j++)
      {
        error_covariance_row = this->Model.BackgroundErrorCovariance(j);
        column(j)
          = this->ObsManager.TLMMltObsOperator(c, error_covariance_row);
      }
    // Computes the column #c of HBH'.
    for (r = 0; r < this->Nobs; r++)
        working_matrix(r, c) =
          this->ObsManager.TLMMltObsOperator(r, column);
  }
```

Data assimilation in Polyphemus

Data assimilation system

Observations

Assimilation algorithms

Data assimilation in Polyphemus

System content

Model

- Polair3D, Chemistry-transport model.

Observations

- Realistic (from stations).
- Synthetic (from perturbations of model reference run).

Assimilation algorithms

- Optimal interpolation (OI)
- Reduced-rank sqare root Kalman filter (RRSQRT)

Observations

class GroundObservationManager

- EMEP network, 151 stations (more network will be available).
- Hourly obs.

class SimObservationManager

- Option : with station.
- Option : with level.
- Option : with location.

Data assimilation in Polyphemus

Algorithms

- Analyze (estimate) model state $\mathbf{x}$ from :
  - Prior information $\mathbf{x}_b$ (also called backgound, usually takes form of previous model run),
  - Observations $\mathbf{y}^o$,
  - and statistics information (background error covariance $\mathbf{B}$, model error covariance $\mathbf{Q}$, and observation error covariance $\mathbf{R}$).
- Sequential algorithms perform analysis (assimilation) simultaneously whenever observations are available, whereas variational algorithms deal with block of observations.

Data assimilation in Polyphemus

Optimal interpolation

- Best Linear Unbiased Estimator (BLUE)
- Analysis formula :

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y}^o - H(\mathbf{x}_b)),$$
$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}.$$

- Flow independent (B fixed), data selection (analysis performed with locally available observations).
- B is implemented using Balgovind correlation function :

$$f(r) = \left(1 + \frac{r}{L}\right) \exp\left(-\frac{r}{L}\right) v_b$$

Data assimilation in Polyphemus

(Extended) Kalman filter

- Flow dependent (background error covariance evolves)
- Initialization :

$$\mathbf{x}_0^a, \mathbf{P}_0^a$$

- Forecast formula :

$$\mathbf{x}_k^f = M_{k-1\to k}(\mathbf{x}_{k-1}^a)$$
$$\mathbf{P}_k^f = \mathbf{M}_{k-1\to k}\mathbf{P}_{k-1}^a\mathbf{M}_{k-1\to k}^T + \mathbf{Q}_{k-1}$$

- Analysis formula :

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k^o - H(\mathbf{x}_k^f)),$$
$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k)^{-1},$$
$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f$$

Data assimilation in Polyphemus

Reduced-rank square root Kalman filter

- Only evolves square root of error covariance, instead of computing the full matrix (RRSQRT, Heemink et al, 2001).
- Initialization :

$$\mathbf{x}_0^a, \mathbf{L}_0^a = [\mathbf{l}_0^{a,1}, \ldots, \mathbf{l}_0^{a,q}]$$

- Forecast formula :

$$\mathbf{x}_k^f = M_{k-1 \to k+1}(\mathbf{x}_{k-1}^a)$$

$$\mathbf{l}_k^{f,i} = \frac{1}{\epsilon}\{M_{k-1\to k}(\mathbf{x}_{k-1}^a + \epsilon \mathbf{l}_{k-1}^{a,i}) - M_{k-1\to k}(\mathbf{x}_{k-1}^a)\}, \quad \epsilon = 1$$

$$\tilde{\mathbf{L}}_k^f = [\mathbf{l}_k^{f,1}, \ldots, \mathbf{l}_k^{f,q}, \mathbf{Q}_{k-1}^{\frac{1}{2}}]$$

$$\mathbf{L}_k^f = \mathbf{\Pi}_k^f \tilde{\mathbf{L}}_k^f$$

Data assimilation in Polyphemus

Reduced-rank square root Kalman filter

- Analysis formula :

$$\mathbf{P}_k^f = \mathbf{L}_k^f \mathbf{L}_k^{f,T}$$
$$\mathbf{K}_k = \mathbf{P}_k^f \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^f \mathbf{H}_k^T + \mathbf{R}_k)^{-1},$$
$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k (\mathbf{y}_k^o - H(\mathbf{x}_k^f)),$$
$$\tilde{\mathbf{L}}_k^a = [(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{L}_k^f, \mathbf{K}_k \mathbf{R}_k^{\frac{1}{2}}]$$
$$\mathbf{L}_k^a = \mathbf{\Pi}_k^a \tilde{\mathbf{L}}_k^a$$

- In this preliminary study, one column of $\mathbf{Q}_{k-1}^{\frac{1}{2}}$ is chosen as :

$$f(r) = \left(1 + \frac{r}{L_Q}\right) \exp\left(-\frac{r}{L_Q}\right) v_Q, \qquad r = \|s - s_o\|$$

s takes all the locations in model grid domain, s_o is the position of one observation location in the model grids.

Data assimilation in Polyphemus

Implementation : class RRSQRTDriver

```
Array<T, 2> ModeMatrix(this->Nstate, Nmode_max, ColumnMajorArray<2>());
ModeMatrix = 0.;    Array<T, 1> state_vector;

/*** Time loop ***/
for (int i = 0; i < this->Model.GetNt(); i++)
  {
    this->Model.InitStep();
    this->OutputSaver.InitStep(this->Model);

    Forecast(state_vector, ModeMatrix);

    // Retrieves observations.
    this->ObsManager.SetDate(this->Model.GetCurrentDate());
    if (this->ObsManager.IsAvailable())
      {
        Analyze(state_vector, ModeMatrix);
        this->Model.SetState(state_vector);
      }
    this->OutputSaver.Save(this->Model);
  }
```

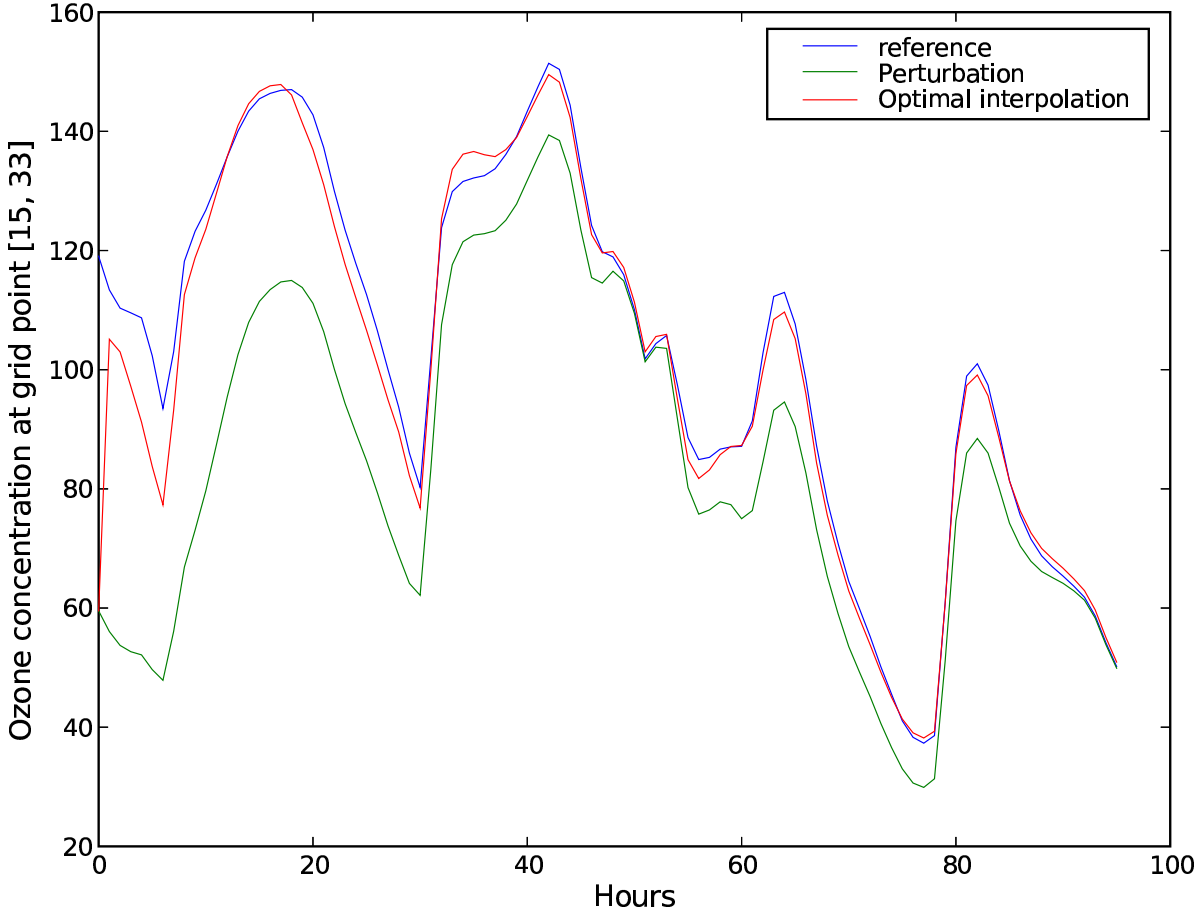
Data assimilation in Polyphemus

In RRSQRTDriver : :Forecast()

```
// Perburbates modes.
if (propagation_option == "finite_difference") {
    for (j = 0; j < Nmode_analyzed; j++) {
        for (i = 0; i < this->Nstate; i++)
          working_vector(i) = old_state(i) +
            finite_difference_perturbation * ModeMatrix(i, j);

        // Perturbed state to be propagated.
        this->Model.SetState(working_vector);

        this->Model.Forward();
        this->Model.GetState(working_vector);
        this->Model.StepBack(concentration);

        // New mode computed on the basis of the perturbation.
        for (i = 0; i < this->Nstate; i++)
          ModeMatrix(i, j) = (working_vector(i) - state_vector(i))
            / finite_difference_perturbation;
      }
  }
```

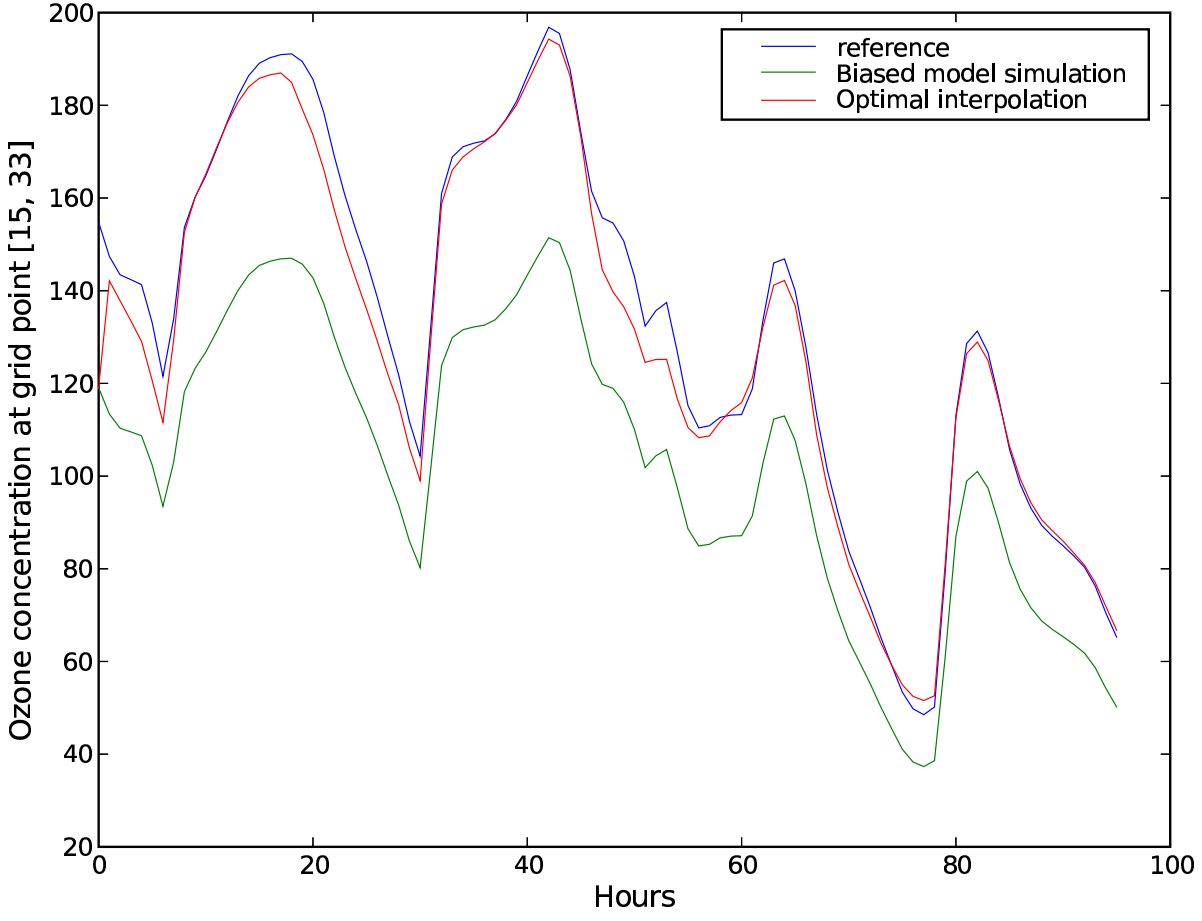
Preliminary results - optimal interpolation

Experiment settings : Starting date 2001/07/05 ; decrease 50% of initial conditions ; observations set to be the reference run ; $\mathbf{R}$ diagonal, variance set to 1 ; Balgovind parameter L set to 2 grid intervals, v_b set to 1.



Preliminary results - optimal interpolation

Experiment settings : biased model (true state 30% more than the model simulation) ; observations set to be the true reference ; other settings same as above.

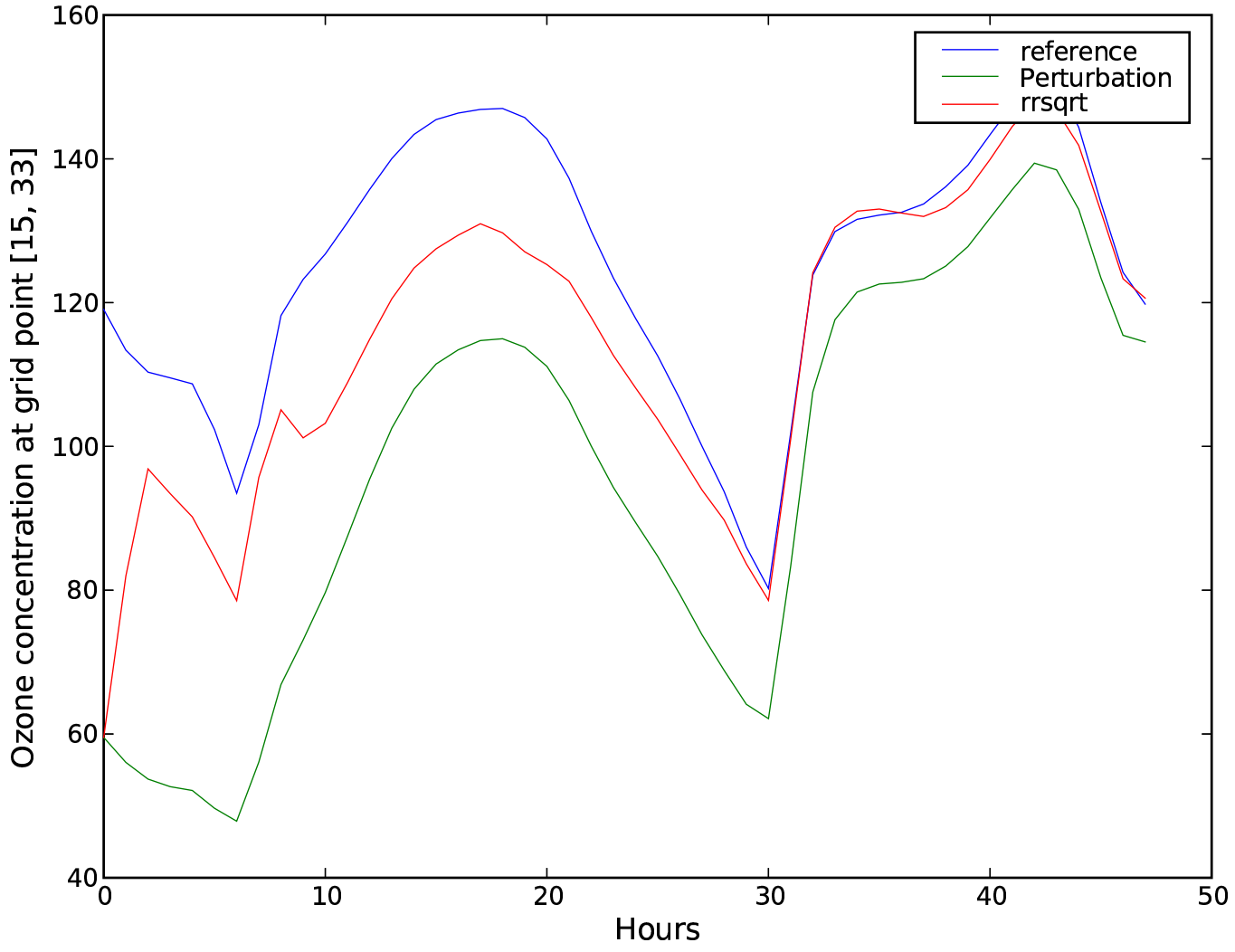


Preliminary results - RRSQRT

Experiment settings :

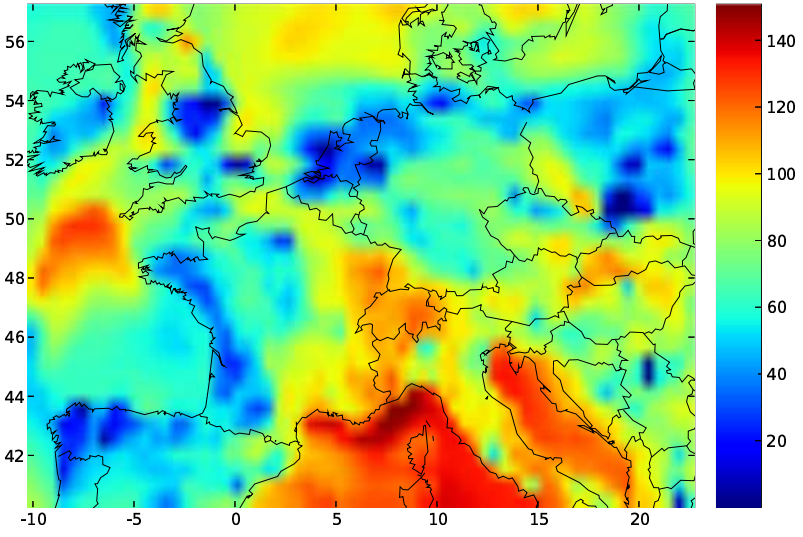
- starting date 2001/07/05 ;
- decrease 50% of initial conditions ;
- observations from reference run ;
- $\mathbf{R}$ diagonal, variance set to 1 ;
- Balgovind parameter L set to 5 grid intervals, v_Q set to 1.,
- RRSQRT parameter q set to 10, 10 columns $\mathbf{Q}^{\frac{1}{2}}$ expected to be added to $\mathbf{L}^f$, 10 columns of $\mathbf{KR}^{\frac{1}{2}}$ expected to be added to $\mathbf{L}^a$.

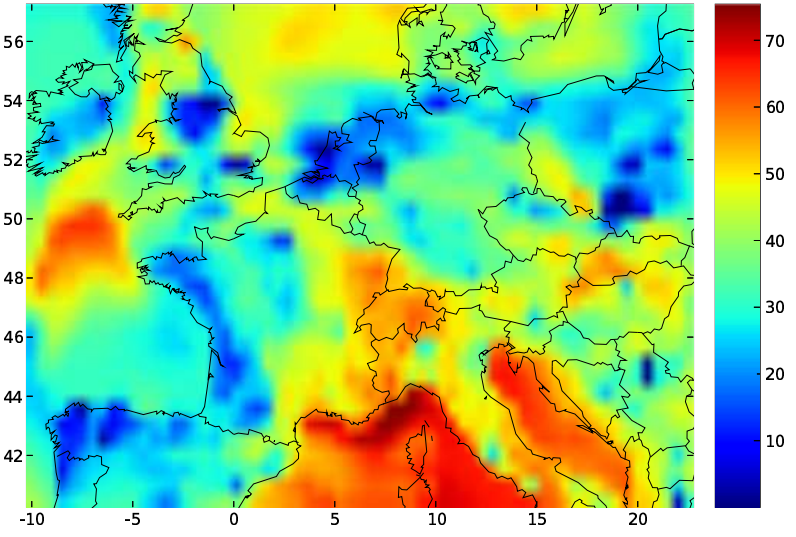
Preliminary results - RRSQRT



Preliminary results - RRSQRT

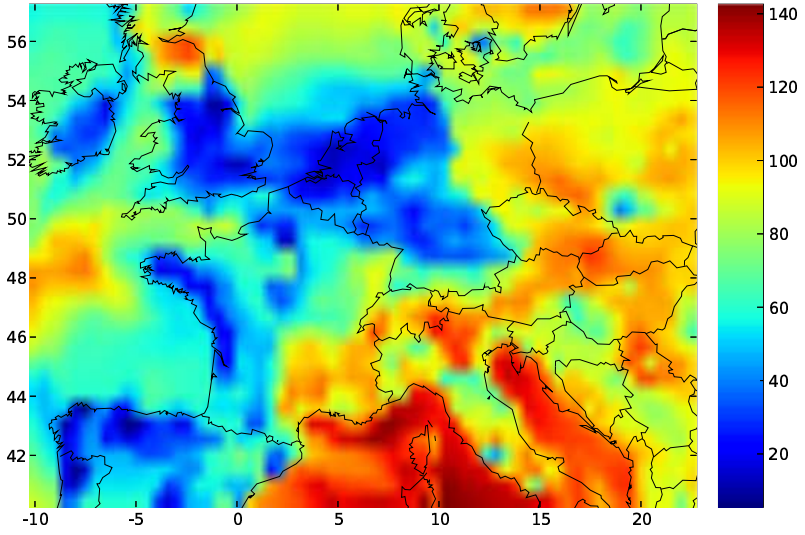
Starting date : 2001/07/05 00H00

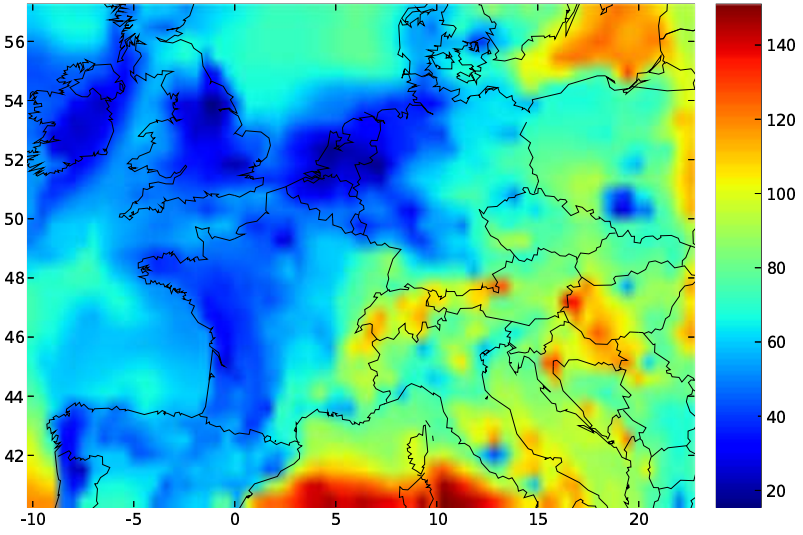




Preliminary results - RRSQRT

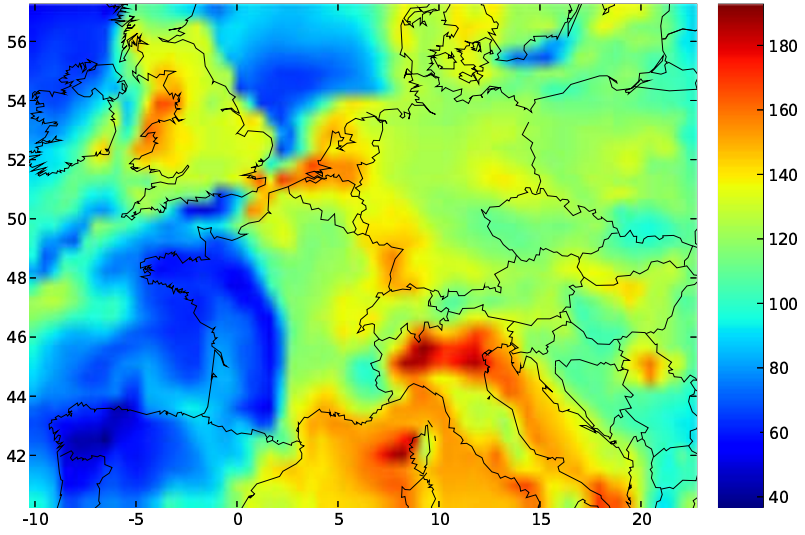
Date : 2001/07/05 06H00

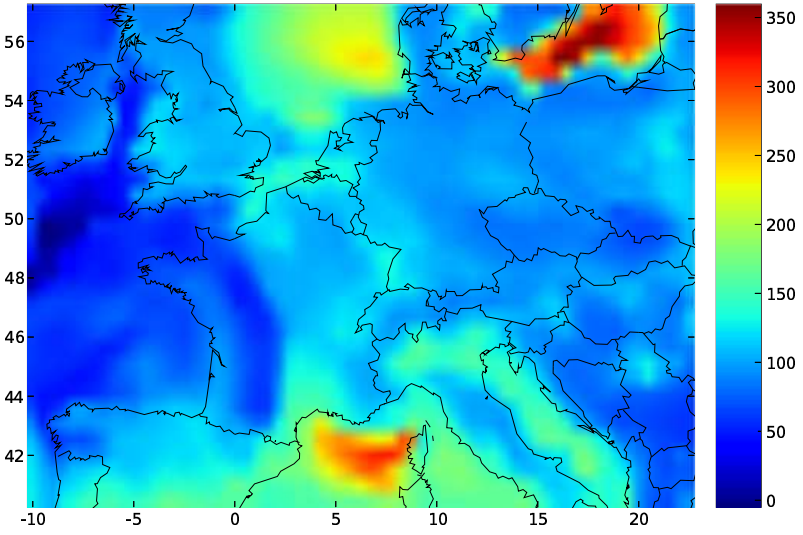




Preliminary results - RRSQRT

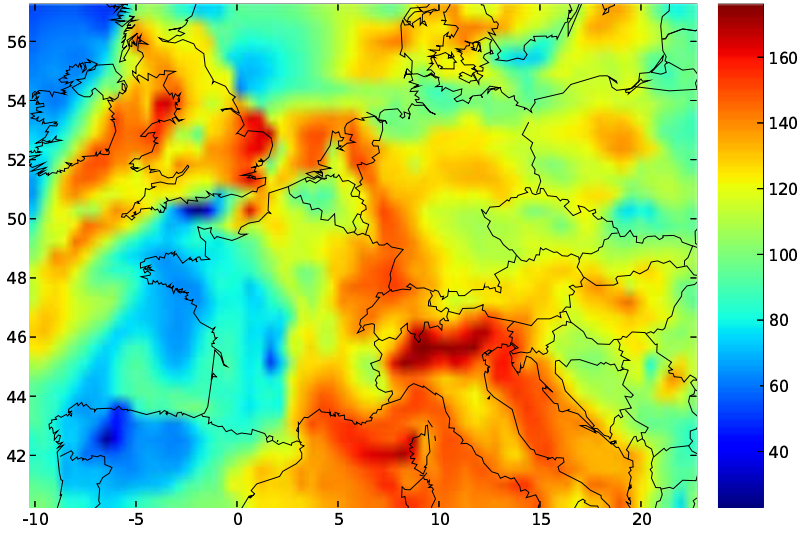
Date : 2001/07/05 12H00

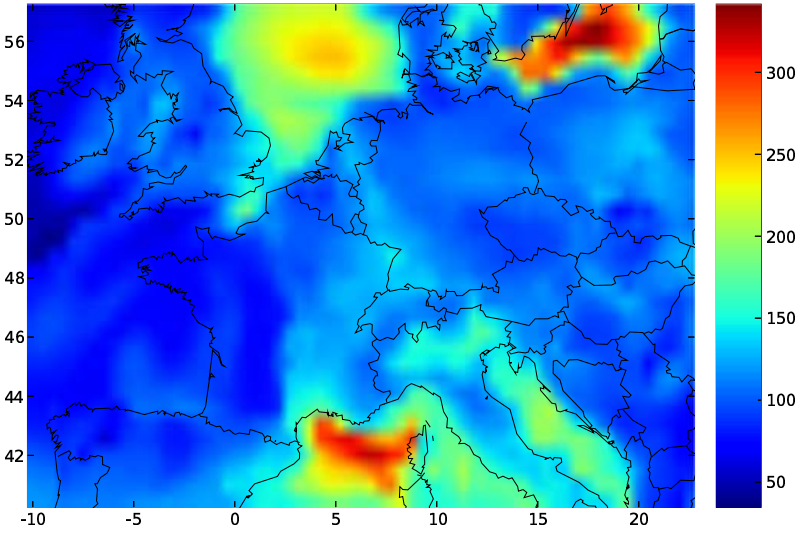




Preliminary results - RRSQRT

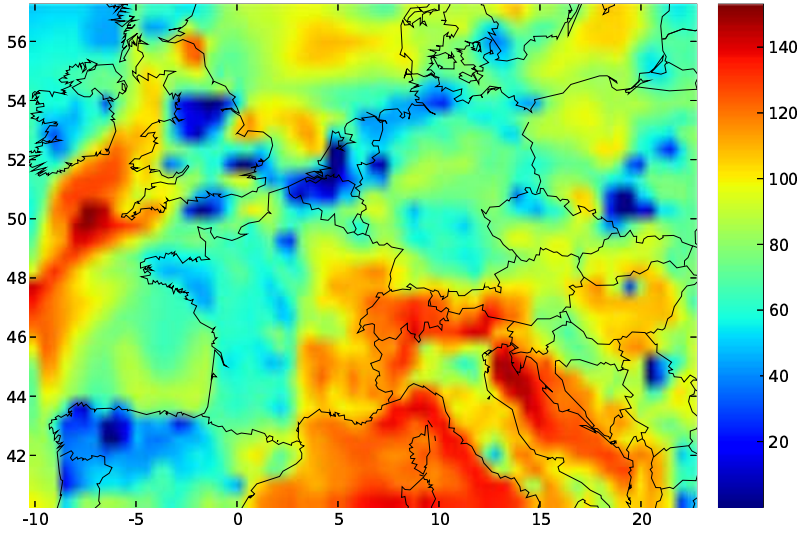
Date : 2001/07/05 18H00

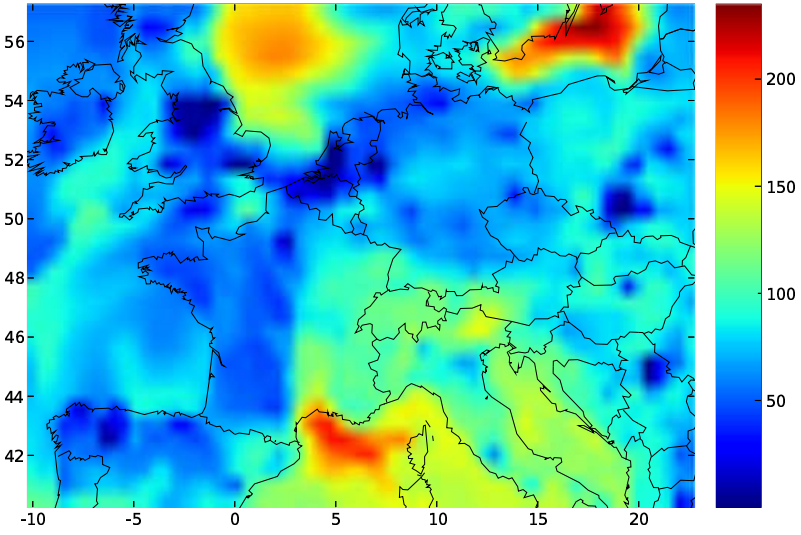




Preliminary results - RRSQRT

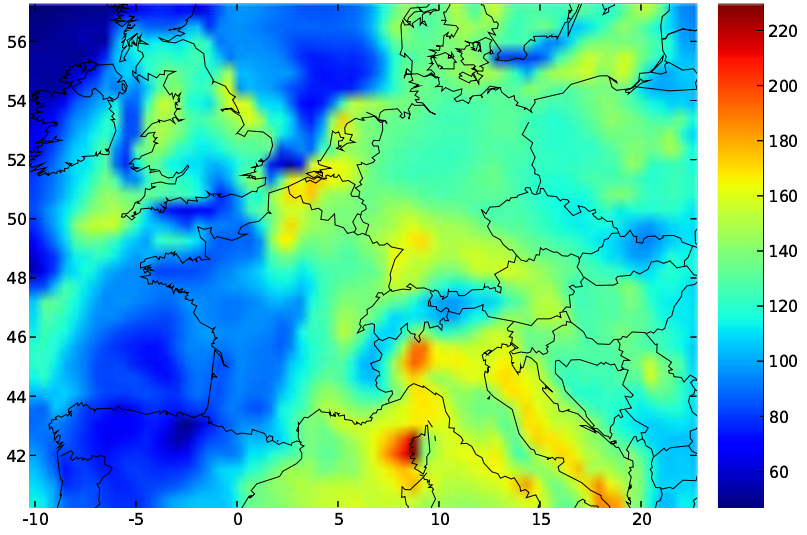
Date : 2001/07/06 00H00

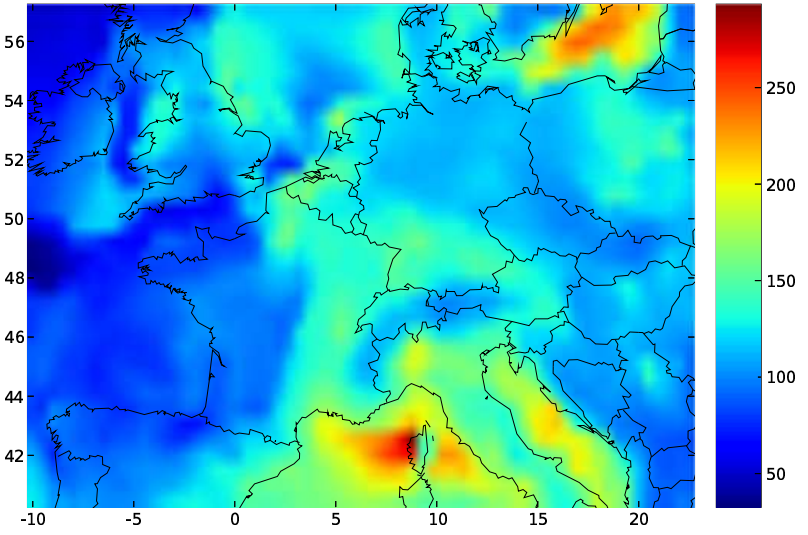




Preliminary results - RRSQRT

Date : 2001/07/06 12H00





Future developments

Tangent linear version of RRSQRT

Verifications of OI and RRSQRT

More realistic modeling of B ***and*** Q

Four-dimensional variational data assimilation