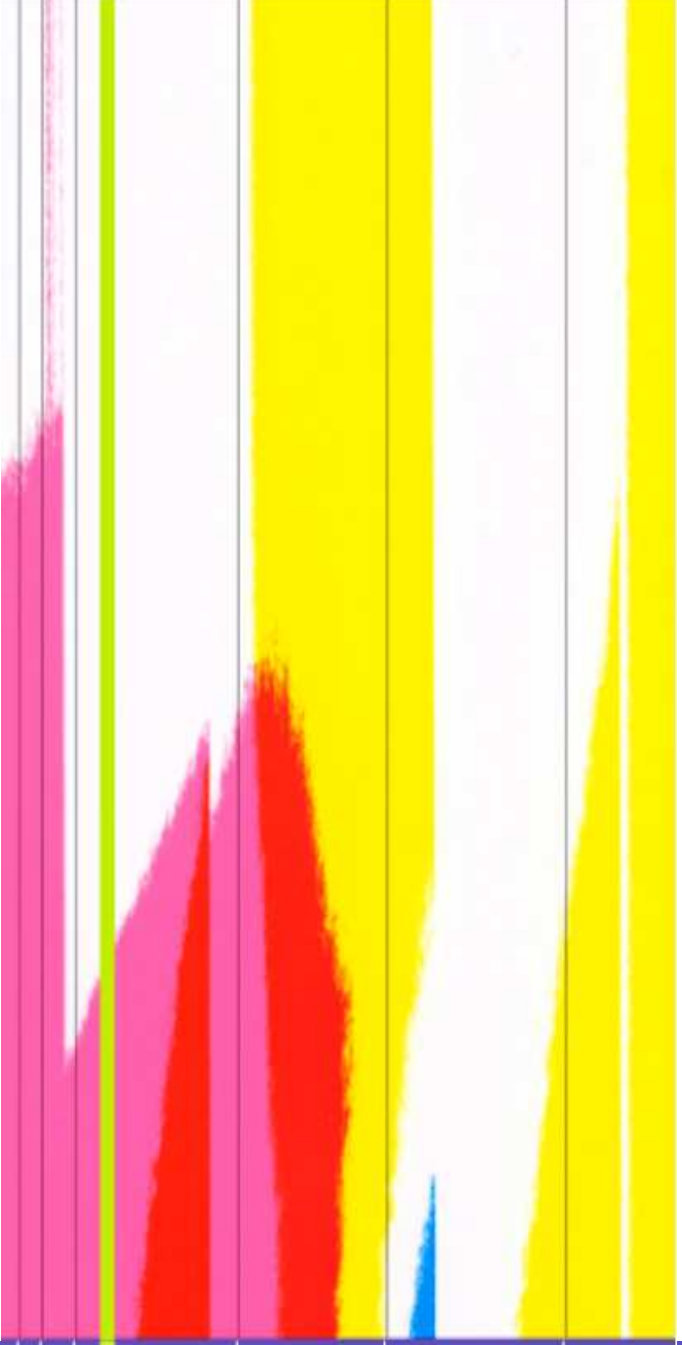


Revue des bibliothèques pour l'assimilation de données et exemple de la bibliothèque Verdandi

Dominique Chapelle[1], Marc Fragu[1], Vivien Mallet[1,2] (orateur), Philippe Moireau[1], Claire Mouton[1,2]

[1]INRIA

[2]CEREA, laboratoire commun École des Ponts ParisTech - EDF R&D, Université Paris-Est

Plan

1. Revue des outils logiciels pour l'assimilation de données
   - DART
   - Oops
   - OpenDA
   - PALM
   - YAO
2. Verdandi
   - Introduction
   - Conception et fonctionnalités
   - Contenu
   - Perspectives

Data Assimilation Research Testbed (DART)

http://www.image.ucar.edu/DAReS/DART/

Objectifs et contenu

- Assimilation fondée sur un ensemble (Kalman d'ensemble)

Contexte de développement et d'application

- National Center for Atmospheric Research (NCAR)
- Licence spécifique autorisant modification et redistribution ; dépôt SVN accessible
- Dernière version stable : avril 2007

Considérations techniques

- Fortran 90
- Script shell pour les appels au modèle (perl et csh/tcsh)

Object-Oriented Prediction System (Oops)

Objectifs et contenu

- 4D-Var incrémental ; probablement EnKF et PSAS
- Assimilation pour le modèle IFS : contraintes opérationnelles (archivage, distribution des calculs)

Contexte de développement et d'application

- ECMWF
- Licence actuellement indéterminée ; peut-être GNU LGPL

Considérations techniques

- Conception objet, centrée sur les données
- C++ et Fortran 90, Python pour les scripts

OpenDA

http://www.openda.org/joomla/

Objectifs et contenu

- Kalman d'ensemble, Kalman RRSQRT, COFFEE
- Visée générale

Contexte de développement et d'application

- Issu de l'université technologique de Delft ; avec deux partenaires, Deltares, VORtech
- Licence GNU LGPL

Considérations techniques

- Java
- Configuration en XLM
- Exécuté par script shell ou via une interface graphique

PALM

http://www.cerfacs.fr/globc/PALM_WEB/

Objectifs et contenu

- Couplage dynamique et parallèle
- Donne un cadre pour l'écriture d'algorithmes d'assimilation

Contexte de développement et d'application

- CERFACS
- Accès gratuit aux organismes de recherche, mais sans support

Considérations techniques

- Nécessite d'introduire des appels PALM dans les codes à coupler
- Mise en place des couplages dans une interface graphique

YAO

http://www.locean-ipsl.upmc.fr/~yao/

Objectifs et contenu

- Support à la génération d'un modèle adjoint ou linéaire tangent
- Représentation des échanges sous forme de graphes

Contexte de développement et d'application

- LOCEAN-IPSL et Centre d'étude et de recherche en informatique du Cnam
- Code fourni sur demande ; passage à la licence CeCILL prévu

Considérations techniques

- Nécessite de décrire les entrées/sorties des modules
- Génère un code C++

Introduction à Verdandi

Bibliothèque d'assimilation de données générique

Objectifs

- Fournir au moins toutes les méthodes d'assimilation classiques
- Proposer des outils pour accompagner la mise en œuvre de l'assimilation
- Permettre l'application à pratiquement tous les problèmes de grande dimension
- Proposer un cadre rigoureux et pérenne pour le développement de méthodes d'assimilation
- Améliorer la diffusion et l'impact des algorithmes d'assimilation de données

Introduction à Verdandi

Bibliothèque d'assimilation de données générique

Utilisateurs potentiels

- Les non-spécialistes, ingénieurs ou chercheurs, qui souhaitent appliquer l'assimilation à leurs problèmes
- Les spécialistes qui peuvent s'appuyer sur un logiciel facilitant le développement, la diffusion et les interactions

Développement actuel

- INRIA (CLIME & MACS)
- Deux ingénieurs à plein temps
  - CLIME : précédemment Claire Mouton, bientôt Kévin Charpentier
  - MACS : Marc Fragu, financé par le projet européen euHeart, `http://www.euheart.eu/`

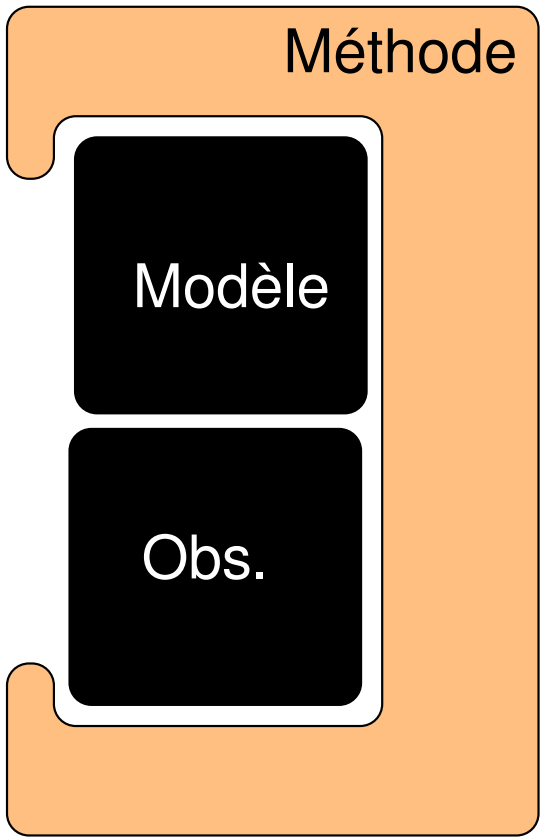
Conception

Conception orientée objet

Langage : C++

Trois objets principaux

- Un objet « modèle »
- Un « gestionnaire des observations »
- Une méthode d'assimilation de données qui pilote les deux premiers



Conception : modèle

Objet « modèle »

- Stocke le vecteur d'état
- Fournit la dynamique, et ses versions linéaire tangente et adjointe
- Définit la variance de l'erreur sur la condition initiale et la variance de l'erreur modèle

Interface C++

- `GetState(x)` où `x` est le vecteur d'état
- `Forward()` ou `ApplyOperator(x)` pour appliquer le modèle
- `ApplyTangentLinearOperator(x)` ou `GetTangentLinearOperator(M)` pour le linéaire tangent
- `GetStateErrorVarianceRow(i, v)` renvoie la i^e ligne de la variance de l'état ; `GetStateErrorVarianceSqrt(L, U)` ($P = LUL^T$)

Conception : modèle

Simulation directe

```
model.Initialize();
while (!model.HasFinished())
{
    model.InitializeStep();
    model.Forward();
}
```

Accès au vecteur d'état

```
Vector x;
model.GetState(x);
[...] // Corrections sur x.
model.SetState(x);
```

Conception : gestion des observations

Objet « gestionnaire d'observations »

- Charge et délivre les observations
- Fournit l'opérateur d'observation et sa linéarisation
- Définit la variance de l'erreur d'observation

Interface C++

- `SetTime(model, t)` où `t` est le temps auquel le gestionnaire doit se positionner
- `GetObservation(y)` où `y` est le vecteur des observations
- `ApplyLinearOperator(x)` et `GetTangentLinearOperator(M)` pour le linéaire tangent
- `GetTangentLinearOperator(),...`
- `GetErrorVariance()` ou `GetErrorVarianceInverse()`

Conception : algorithme d'assimilation

Objet « méthode d'assimilation »

- Encapsule le modèle et le gestionnaire d'observations
- Communique directement avec le modèle et le gestionnaire d'observations
- Effectue les calculs sur la base des variables fournies par le modèle et le gestionnaire d'observations
- Envoie des signaux que tous les objets peuvent recevoir (par exemple, la disponibilité d'une analyse)

Conception : algorithme d'assimilation

```
model.Initialize();
observation_manager.Initialize(model);

while (!model.HasFinished())
{
    model.Forward();
    observation_manager.SetTime(model.GetTime());
    if (observation_manager.HasObservation())
    {
        model.GetState(x);
        observation_manager.GetInnovation(x, d);
        x = x + K * d;
        model.SetState(x);
    }
}
```

Conception : algorithme d'assimilation

Type des variables manipulées

- Plusieurs variables de natures différentes sont manipulées : vecteur d'état, vecteur d'observations, matrices de covariance, lignes des matrices de covariance, opérateur linéaire tangent, matrices intermédiaires de différentes dimensions, ...
- Plusieurs types de stockage peuvent être utilisés : vecteurs, vecteurs distribués, matrices pleines, matrices creuses, matrices par blocs, ...
- Disponibilité des structures de la bibliothèque d'algèbre linéaire Seldon (plein, creux, par bloc)
- *Le choix des types est délégué au modèle et au gestionnaire d'observations*, donc à l'utilisateur

Fonctionnalités

Interface Python générée par SWIG

```
>>> import verdandi
>>> method = verdandi.AssimilationMethod('configuration_file.lua')
>>> method.Initialize()
>>> model = method.GetModel()
>>> print model.GetTime()
    0.0
>>> method.Forward()
>>> print model.GetTime()
    1.0
>>> model.Forward()
>>> print model.GetTime()
    2.0
>>> for i in range(998):
>>>   model.Forward()
>>> print model.GetTime()
    1000.0
```

Fonctionnalités

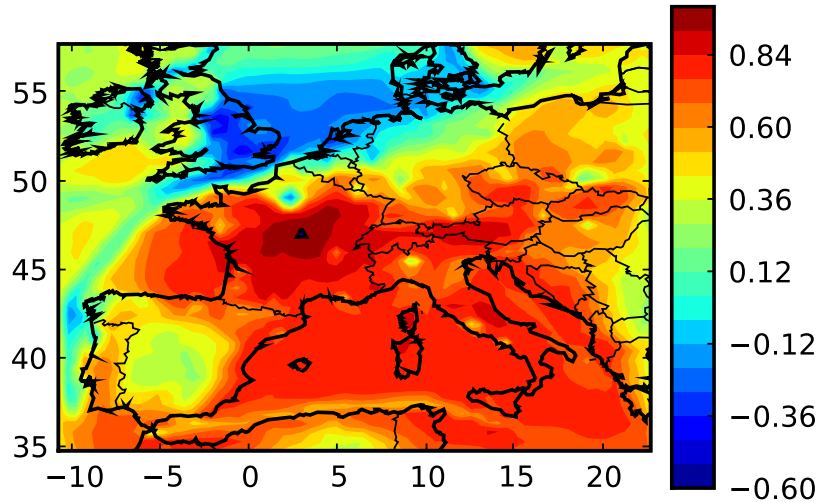
Interface Python générée par SWIG

```
>>> state_vector = seldon.VectorDouble()
>>> model.GetState(state_vector)
>>> state_vector.Print()
    1.0167881998005845468 1.0150381428637218484 ...
>>> method.Analyze()
>>> model.GetState(state_vector)
>>> state_vector.Print()
    1.0167882075200211922 1.0150381458769643928 ...
>>> state_vector[0] = 0.5
>>> model.SetState(state_vector)
>>> model.GetState(state_vector)
>>> state_vector.Print()
    0.5 1.0150381458769643928 ...
>>> plot(state_vector)
>>> print array(state_vector).mean()
    0.99768
```

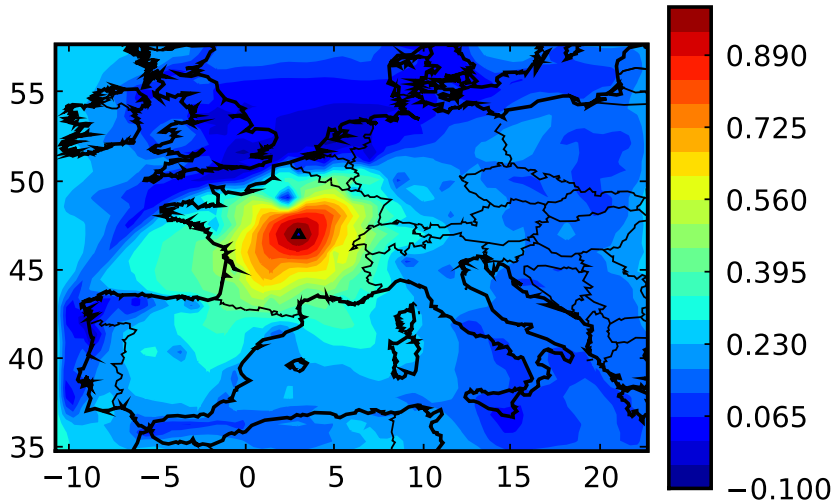
Fonctionnalités

Perturbation de champs

Illustration sur une simulation Monte Carlo



Corrélation sur le champ d'ozone (sortie), après perturbations homogènes des entrées.



Corrélation sur le champ d'ozone (sortie), après perturbations spatialisées des entrées.

Crédit : Anne Tilloy (INRIA, CEREA)

Fonctionnalités

Configuration avec fichiers Lua

- Syntaxe très simple et flexible
- Langage de script
  - Permet un grand nombre d'opérations (concaténation de chaînes, définition de tableaux de données, calculs numériques, . . .)
  - Permet d'inclure un fichier de configuration dans un autre, et d'écraser une partie de la configuration
  - Permet les appels systèmes (création du répertoire de sortie, . . .)
- Les valeurs effectivement lues peuvent être sauvegardées sur disque

Divers

- Agrégation temporelle des observations
- Fichiers de log
- Gestion des sauvegardes
- Compilation SCons

Contenu de Verdandi

Algorithmes

- Interpolation optimale
- Kalman étendu, et une version réduite (racine)
- Kalman sans parfum, et une version réduite (cf. la présentation de Philippe Moireau)
- Filtre minimax réduit (cf. le poster correspondant, Mallet & Zhuk)

Gestionnaires d'observation

Un gestionnaire linéaire, et un gestionnaire permettant le passage d'une grille à un réseau d'observation

Modèles d'exemple

Barre encastrée, Lorenz, modèle quadratique, modèle de Saint-Venant

Ressources disponibles

Documents

- Guide d'utilisation et documentation de référence intégrés
- Conventions d'implémentation
- Conception : document d'introduction et de discussion ; revue des bibliothèques d'algèbre linéaire

Code source

- Version « bêta » (0.8) disponible sous GNU LGPL
  `http://verdandi.gforge.inria.fr/`
- Miroir du dépôt de référence Git
  `http://gitorious.org/verdandi/`
- Liste d'aide `verdandi-help@lists.gforge.inria.fr`
- Report de bugs et propositions d'amélioration
  `http://gforge.inria.fr/projects/verdandi/`

Perspectives

Quelques développements en cours ou prévus

- Kalman d'ensemble, gestion des nombres aléatoires
- 4D-Var, algorithmes d'optimisation
- Meilleure gestion de la parallélisation
- Prévision d'ensemble par agrégation séquentielle

Version 1.0

- Première version destinée à être distribuée largement
- Sortie au plus tard en juin

Point d'entrée

`http://verdandi.gforge.inria.fr/`