

YAO

a Software for Variational Data Assimilation using Numerical Models

Luigi NARDI

Charles SORROR Fouad BADRAN Sylvie THIRIA



LOCEAN : Laboratoire d'Océanographie et du Climat
Expérimentation et Approches Numériques.
CNRS / IRD / UPMC / MNHN / IPSL
CEDRIC : Centre d'Etude et De Recherche en
Informatique du Cnam

Paris France



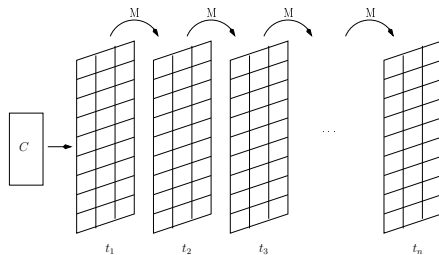
Outline

- 1 Data assimilation
- 2 Modular graph and algorithms
- 3 YAO architecture
- 4 Perspective and conclusion

- 1 Data assimilation
- 2 Modular graph and algorithms
- 3 YAO architecture
- 4 Perspective and conclusion

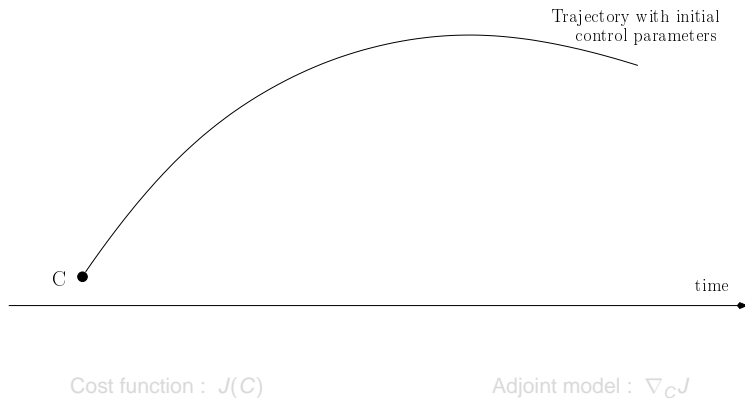
Numerical direct model

- M : direct model between two time steps t_i, t_{i+1}
- C : control parameters

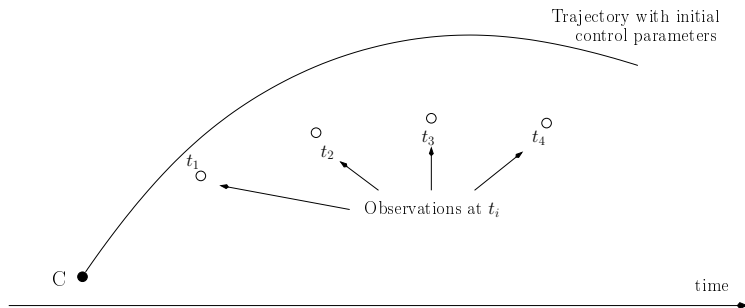


Models are used to forecast or analyse the evolution of phenomena but. . . numerical models are not perfects !

Variational data assimilation idea



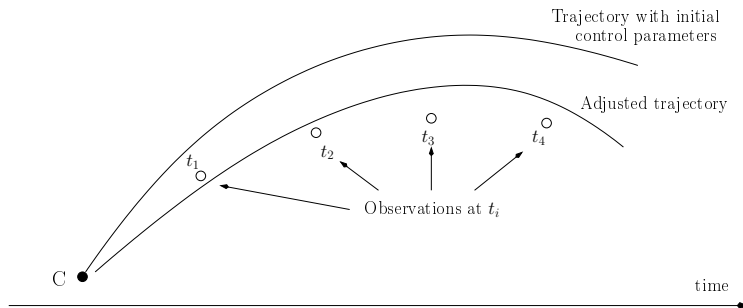
Variational data assimilation idea



Cost function : $J(C)$

Adjoint model : $\nabla_C J$

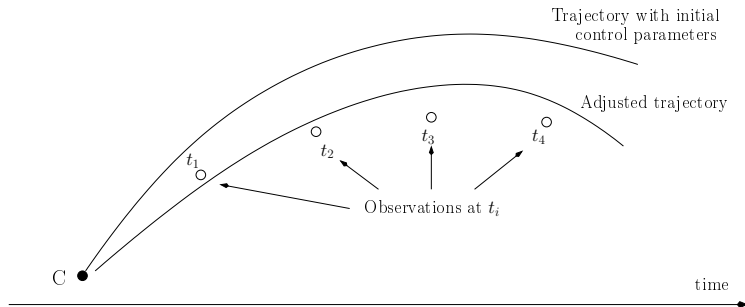
Variational data assimilation idea



Cost function : $J(C)$

Adjoint model : $\nabla_C J$

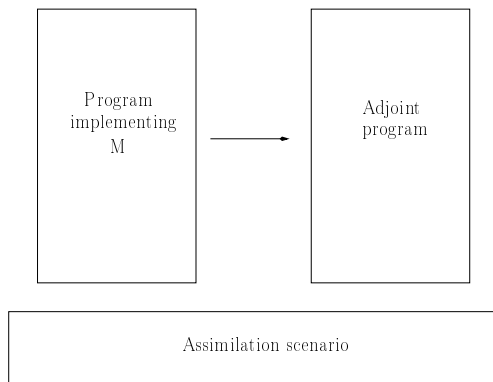
Variational data assimilation idea



Cost function : $J(C)$

Adjoint model : $\nabla_C J$

Programming codes



Write the adjoint program is very time expensive

Assimilation software

- Generation of the adjoint model by automatic differentiation : **Tapenade, OpenAD, Taf**
- Schedule the various calculations according to a scenario and to the chosen minimization method : **Palm**

YAO

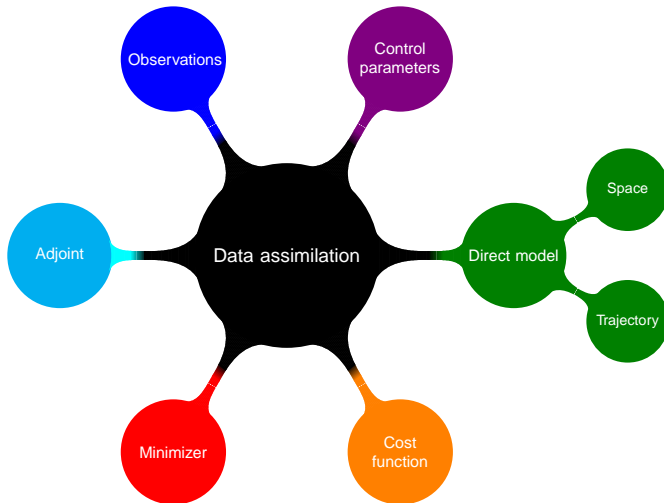
concerns these two tasks and deals with the two problems simultaneously

Assimilation software

- Generation of the adjoint model by automatic differentiation : Tapenade, OpenAD, Taf
- Schedule the various calculations according to a scenario and to the chosen minimization method : Palm

YAO
concerns these two tasks and deals with the two problems
simultaneously

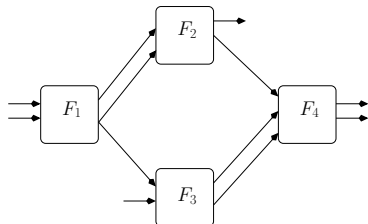
Data assimilation components



- 1 Data assimilation
- 2 Modular graph and algorithms**
- 3 YAO architecture
- 4 Perspective and conclusion

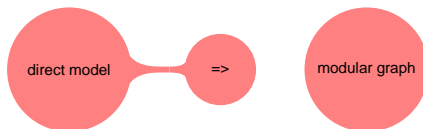
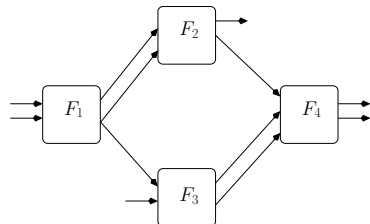
Modular graph

- *modul* : function representing a computation entity
- *basic connection* : data transmission between modules



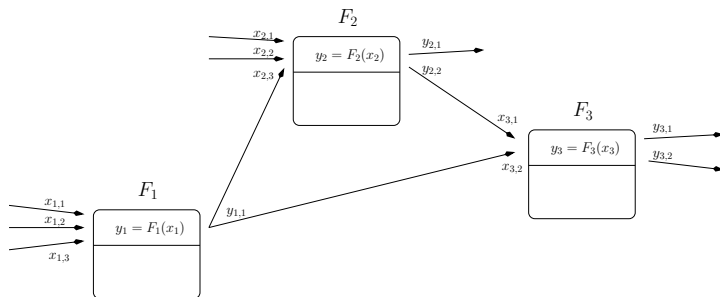
Modular graph

- *modul* : function representing a computation entity
- *basic connection* : data transmission between modules



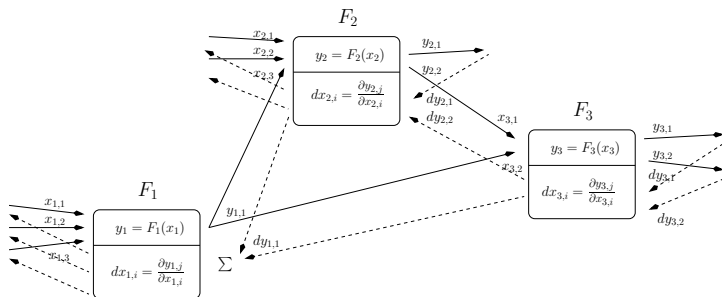
Direct model

Passing through the graph in a topological order we calculate the direct model : **forward algorithm**



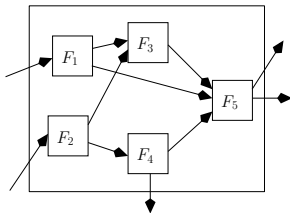
Adjoint model

Passing through the graph in a reverse topological order we calculate the adjoint model : **backward algorithm**



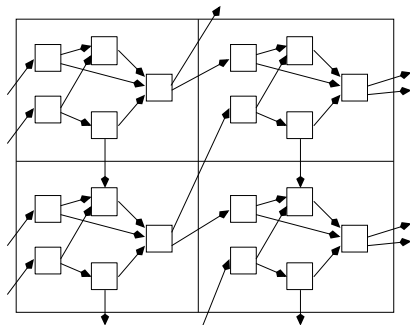
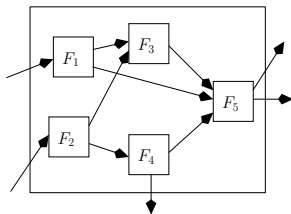
Graph replication

The same graph is repeated for each grid point and time step



Graph replication

The same graph is repeated for each grid point and time step



The shallow water numerical model

Dynamic variables :

$$u_{ijt} = \hat{u}_{ijt-2} + 2\Delta t \left(\frac{-g^*}{\Delta x} [h_{i+1jt-1} - h_{ijt-1}] + \frac{f}{4} [v_{ijt-1} + v_{ij+1t-1} + v_{i+1jt-1} + v_{i+1j+1t-1}] - \gamma \cdot \hat{u}_{ijt-2} \right)$$

$$v_{ijt} = \hat{v}_{ijt-2} + 2\Delta t \left(\frac{-g^*}{\Delta y} [h_{ijt-1} - h_{ij-1t-1}] - \frac{f}{4} [u_{i-1j-1t-1} + u_{i-1jt-1} + u_{ij-1t-1} + u_{ijt-1}] - \gamma \cdot \hat{v}_{ijt-2} \right)$$

$$h_{ijt} = \hat{h}_{ijt-2} - 2\Delta t \cdot H \left(\frac{u_{ijt-1} - u_{i-1jt-1}}{\Delta x} + \frac{v_{ij+1t-1} - v_{ijt-1}}{\Delta y} \right)$$

Asselin filter :

$$\hat{u}_{ijt} = u_{ijt-1} + \alpha(\hat{u}_{ijt-1} - 2u_{ijt-1} + u_{ijt})$$

$$\hat{v}_{ijt} = v_{ijt-1} + \alpha(\hat{v}_{ijt-1} - 2v_{ijt-1} + v_{ijt})$$

$$\hat{h}_{ijt} = h_{ijt-1} + \alpha(\hat{h}_{ijt-1} - 2h_{ijt-1} + h_{ijt})$$

u,v : horizontal velocities

h : height of the water

Description

```
trajectory 100
space 50 50
```

```
modul  $\hat{H}$  input 3 output 1
modul  $\hat{U}$  input 3 output 1
modul  $\hat{V}$  input 3 output 1
modul H input 5 output 1
modul U input 7 output 1
modul V input 7 output 1
```

```
connection  $\hat{H}$  from  $\hat{H}$  i j t-1
connection  $\hat{H}$  from H i j t-1
connection  $\hat{H}$  from H i j t
connection  $\hat{U}$  from  $\hat{U}$  i j t-1
connection  $\hat{U}$  from U i j t-1
connection  $\hat{U}$  from U i j t
connection  $\hat{V}$  from  $\hat{V}$  i j t-1
connection  $\hat{V}$  from V i j t-1
connection  $\hat{V}$  from V i j t
connection H from  $\hat{H}$  i j t-1
connection H from U i j t-1
```

```
...
...
```

```
order i j
      H U V  $\hat{H}$   $\hat{U}$   $\hat{V}$ 
forder
```

Description

```
trajectory 100
space 50 50
```

```
modul  $\hat{H}$  input 3 output 1
modul  $\hat{U}$  input 3 output 1
modul  $\hat{V}$  input 3 output 1
modul H input 5 output 1
modul U input 7 output 1
modul V input 7 output 1
```

```
connection  $\hat{H}$  from  $\hat{H}$  i j t-1
connection  $\hat{H}$  from H i j t-1
connection  $\hat{H}$  from H i j t
connection  $\hat{U}$  from  $\hat{U}$  i j t-1
connection  $\hat{U}$  from U i j t-1
connection  $\hat{U}$  from U i j t
connection  $\hat{V}$  from  $\hat{V}$  i j t-1
connection  $\hat{V}$  from V i j t-1
connection  $\hat{V}$  from V i j t
connection H from  $\hat{H}$  i j t-1
connection H from U i j t-1
...
...
```

```
order i j
      H U V  $\hat{H}$   $\hat{U}$   $\hat{V}$ 
forder
```

Description

```
trajectory 100
space 50 50
```

```
modul  $\hat{H}$  input 3 output 1
modul  $\hat{U}$  input 3 output 1
modul  $\hat{V}$  input 3 output 1
modul H input 5 output 1
modul U input 7 output 1
modul V input 7 output 1
```

```
connection  $\hat{H}$  from  $\hat{H}$  i j t-1
connection  $\hat{H}$  from H i j t-1
connection  $\hat{H}$  from H i j t
connection  $\hat{U}$  from  $\hat{U}$  i j t-1
connection  $\hat{U}$  from U i j t-1
connection  $\hat{U}$  from U i j t
connection  $\hat{V}$  from  $\hat{V}$  i j t-1
connection  $\hat{V}$  from V i j t-1
connection  $\hat{V}$  from V i j t
connection H from  $\hat{H}$  i j t-1
connection H from U i j t-1
...
...
```

```
order i j
      H U V  $\hat{H}$   $\hat{U}$   $\hat{V}$ 
forder
```

Description

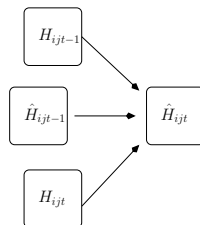
```
trajectory 100
space 50 50
```

```
modul  $\hat{h}$  input 3 output 1
modul  $\hat{U}$  input 3 output 1
modul  $\hat{V}$  input 3 output 1
modul H input 5 output 1
modul U input 7 output 1
modul V input 7 output 1
```

```
connection  $\hat{H}$  from  $\hat{H}$  i j t-1
connection  $\hat{H}$  from H i j t-1
connection  $\hat{H}$  from H i j t
connection  $\hat{U}$  from  $\hat{U}$  i j t-1
connection  $\hat{U}$  from U i j t-1
connection  $\hat{U}$  from U i j t
connection  $\hat{V}$  from  $\hat{V}$  i j t-1
connection  $\hat{V}$  from V i j t-1
connection  $\hat{V}$  from V i j t
connection H from  $\hat{H}$  i j t-1
connection H from U i j t-1
...
...
```

```
order i j
      H U V  $\hat{H}$   $\hat{U}$   $\hat{V}$ 
forder
```

$$\hat{h}_{ijt} = h_{ijt-1} + \alpha(\hat{h}_{ijt-1} - 2h_{ijt-1} + h_{ijt})$$



Description

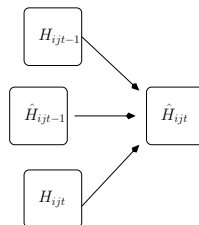
```
trajectory 100
space 50 50
```

```
modul  $\hat{h}$  input 3 output 1
modul  $\hat{u}$  input 3 output 1
modul  $\hat{v}$  input 3 output 1
modul H input 5 output 1
modul U input 7 output 1
modul V input 7 output 1
```

```
connection  $\hat{h}$  from  $\hat{h}$  i j t-1
connection  $\hat{h}$  from H i j t-1
connection  $\hat{h}$  from H i j t
connection  $\hat{u}$  from  $\hat{u}$  i j t-1
connection  $\hat{u}$  from U i j t-1
connection  $\hat{u}$  from U i j t
connection  $\hat{v}$  from  $\hat{v}$  i j t-1
connection  $\hat{v}$  from V i j t-1
connection  $\hat{v}$  from V i j t
connection H from  $\hat{h}$  i j t-1
connection H from U i j t-1
...
...
```

```
order i j
      H U V  $\hat{h}$   $\hat{u}$   $\hat{v}$ 
forder
```

$$\hat{h}_{ijt} = h_{ijt-1} + \alpha(\hat{h}_{ijt-1} - 2h_{ijt-1} + h_{ijt})$$



Real applications

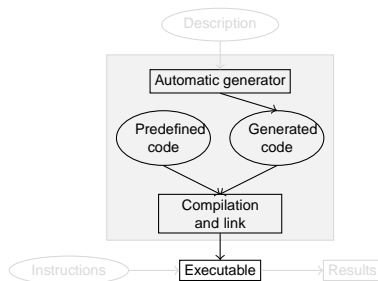
Several variational data assimilation applications have been developed :

- **Ocean color** : variational inversion of multi-spectral satellite ocean color measurements for the restitution of the chlorophyll-a
- **Marine acoustics** : variational inversion of sound speed profile and retrieval of geoacoustic parameters (celerity, density, attenuation, ...)
- **PISCES** : ocean color variational data assimilation in a biogeochemical model
- **NEMO** : adjoint model of the GYRE configuration of NEMO (Nucleus for European Modelling of the Ocean)
- **ISBA** : variational data assimilation in the hydrology model of interface ground - vegetation - atmosphere

- 1 Data assimilation
- 2 Modular graph and algorithms
- 3 YAO architecture**
- 4 Perspective and conclusion

YAO architecture

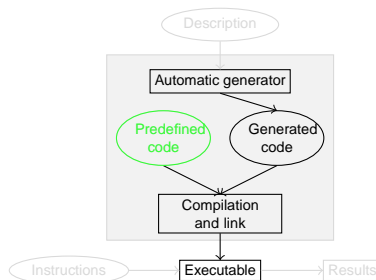
YAO is composed by an assimilation predefined code and a model dependent code



YAO architecture

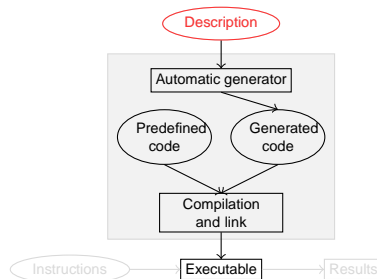
YAO is composed by an assimilation predefined code and a model dependent code

Common data
assimilation code
provided by YAO



YAO architecture

YAO is composed by an assimilation predefined code and a model dependent code

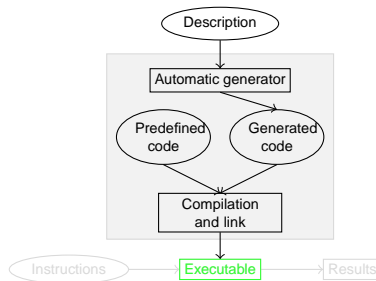


Specific application code provided by the user

YAO architecture

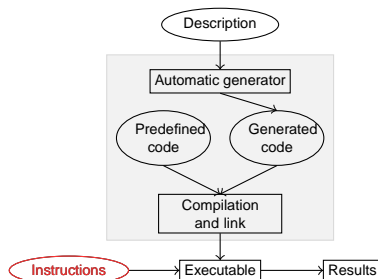
YAO is composed by an assimilation predefined code and a model dependent code

Generation and
compilation
made by YAO



YAO architecture

YAO is composed by an assimilation predefined code and a model dependent code



Assimilation
scenario defined
by the user

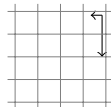
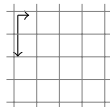
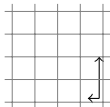
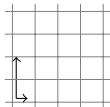
- 1 Data assimilation
- 2 Modular graph and algorithms
- 3 YAO architecture
- 4 Perspective and conclusion**

Computation order verification/generation

Verification objective : understand if there are incoherences between the order defined by the user and the modular graph

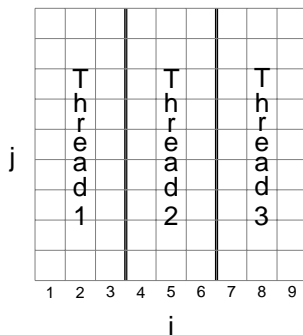
```
order i j
  M1 M2 M3 M4
forder
```

```
order i
  M1 M2
  order j
    M3 M4
  forder
    order j
      M5
    forder
  forder
```



Generation objective : generate the order automatically analyzing the modular graph

Automatic code parallelization



- Operate a domain decomposition
- The modular graph allows to understand automatically where there are dependences that prevent the parallelization. The analysis isolates the sequential parts

New features

- the adjoint computation needs a huge memory allocation.
Checkpointing : the idea is to store less data and to recompute from a snapshot
- graphics interface to help the user to conceive the modular graph

Conclusions

YAO is a software for variational data assimilation



Minimal programming
effort required

The adjoint model is
automatically provided

Flexible : if a model is upgraded
little modifications are required

Conclusions

YAO is a software for variational data assimilation




Minimal programming
effort required

The adjoint model is
automatically provided

Flexible : if a model is upgraded
little modifications are required

Conclusions

YAO is a software for variational data assimilation




Minimal programming
effort required

The adjoint model is
automatically provided

Flexible : if a model is upgraded
little modifications are required

Conclusions

YAO is a software for variational data assimilation

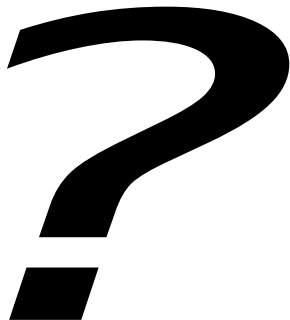


Minimal programming
effort required

The adjoint model is
automatically provided

Flexible : if a model is upgraded
little modifications are required

Questions



YAO home page : <http://www.locean-ipsl.upmc.fr/~yao>
YAO group email : lnalod@locean-ipsl.upmc.fr