Automatic Differentiation: A Tool For Data Assimilation in Oceanography

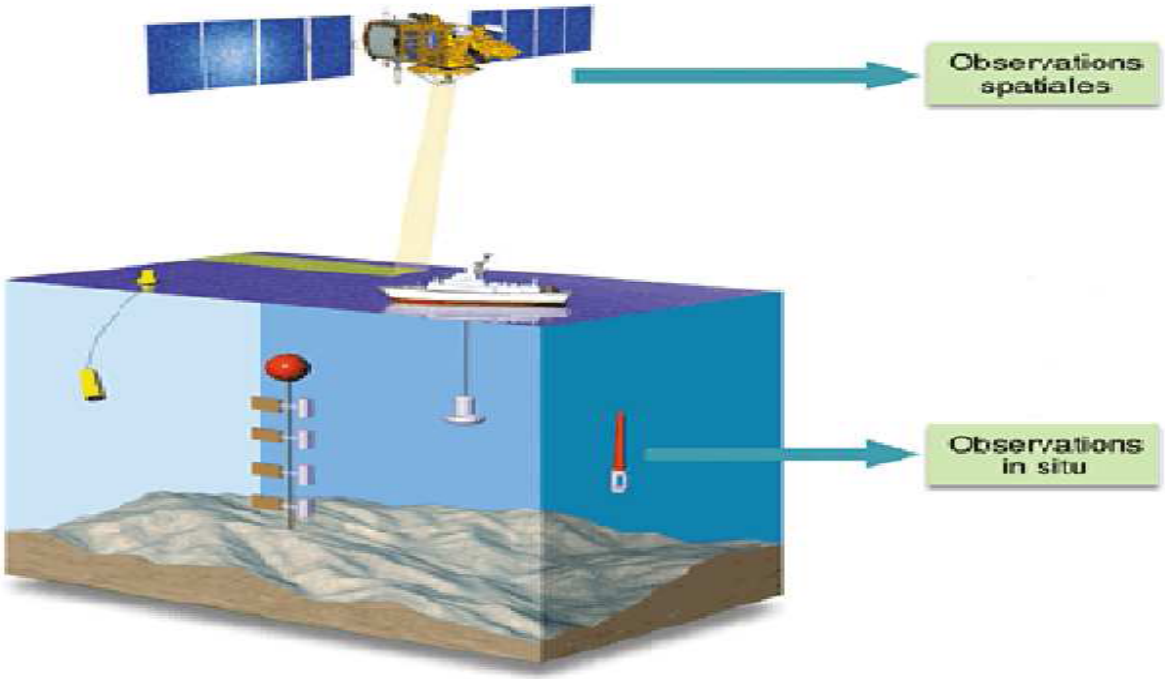
Moulay HICHAM TBER*

Projet TROPICS, INRIA, 2004 route des Lucioles Sophia Antipolis
hicham.tber@sop-inria.fr

Plan

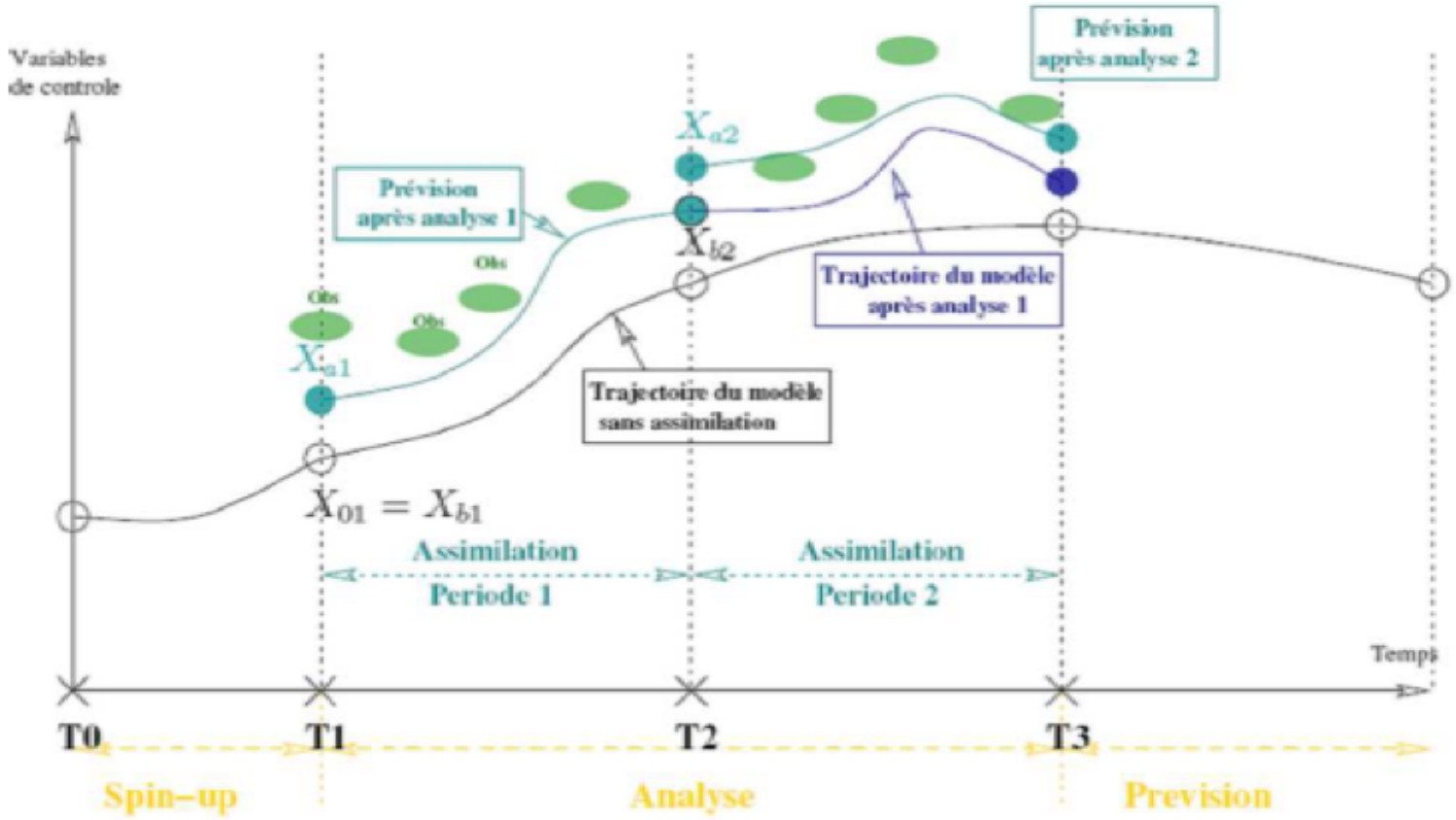
Data Assimilation in Oceanography

- Model $\mathcal{M}$ describing the distribution and evolution in space and time of the characteristics of the sea- State Variables Y-(velocity, temperature, pressure, ..)
- $\mathcal{M}$ depends, among others, on the initial conditions $Y_0 = X$.
- Observations : In-Situ, Spatial



- Data assimilation = estimating initial conditions (in our context)

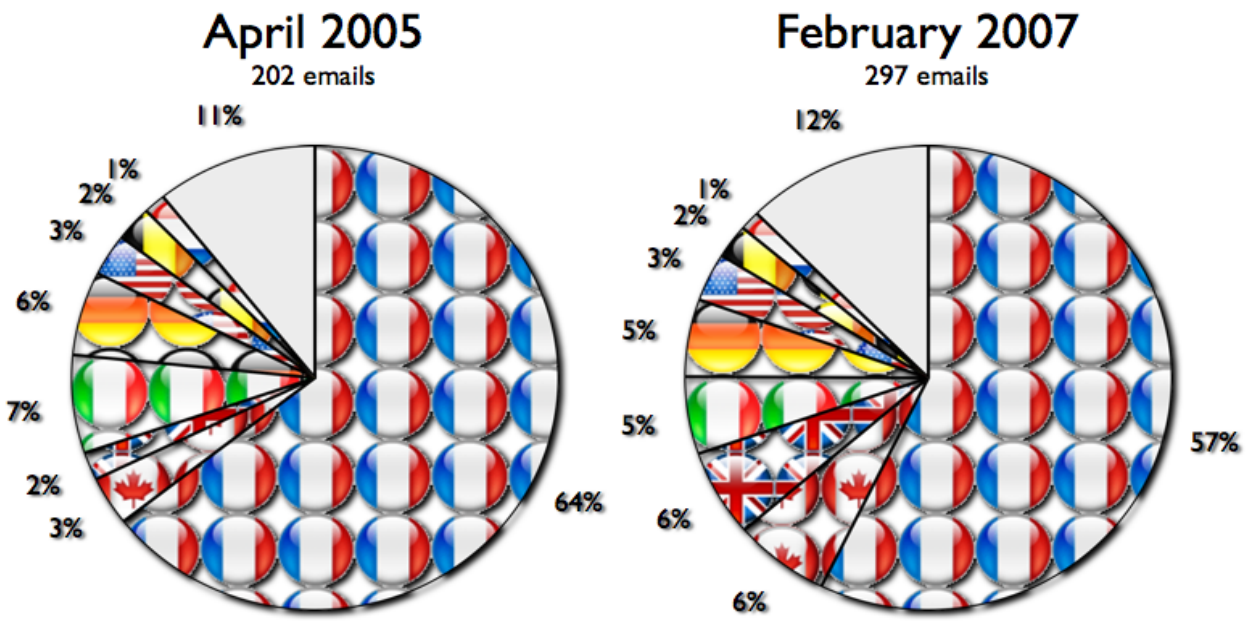
- Observations guide the model on a realistic trajectory
- Model provide a dynamic space-time extrapolation/intrapolation of the observations.



Plan

Direct Model: OPA/NEMO

- Developed at LODYC-LOCEAN-Paris VI
- Global ocean circulation model
- The only european global ocean circulation model used in research and for operational oceanographic purposes (according to IPSL).
- Number and repartition of NEMO users (2007 users meeting statistics)



Primitive Equations

- Navier-Stokes equations (mass conservation and momentum conservation, including the Coriolis force)
- State equation for water density and heat equation
- Boussinesq and hydrostatic approximations

$$\begin{cases} \partial_t u + \mathbf{U}\nabla u - fv + \frac{1}{\rho_0}\partial_x p = D_u \\ \partial_t v + \mathbf{U}\nabla v - fu + \frac{1}{\rho_0}\partial_y p = D_v \\ \partial_t T + \mathbf{U}\nabla T = D_T \; \partial_t S + \mathbf{U}\nabla S = D_S \\ \partial_z p = -\rho g \\ div\mathbf{U} = 0 \\ \rho = \rho(T, S, p) \end{cases}$$

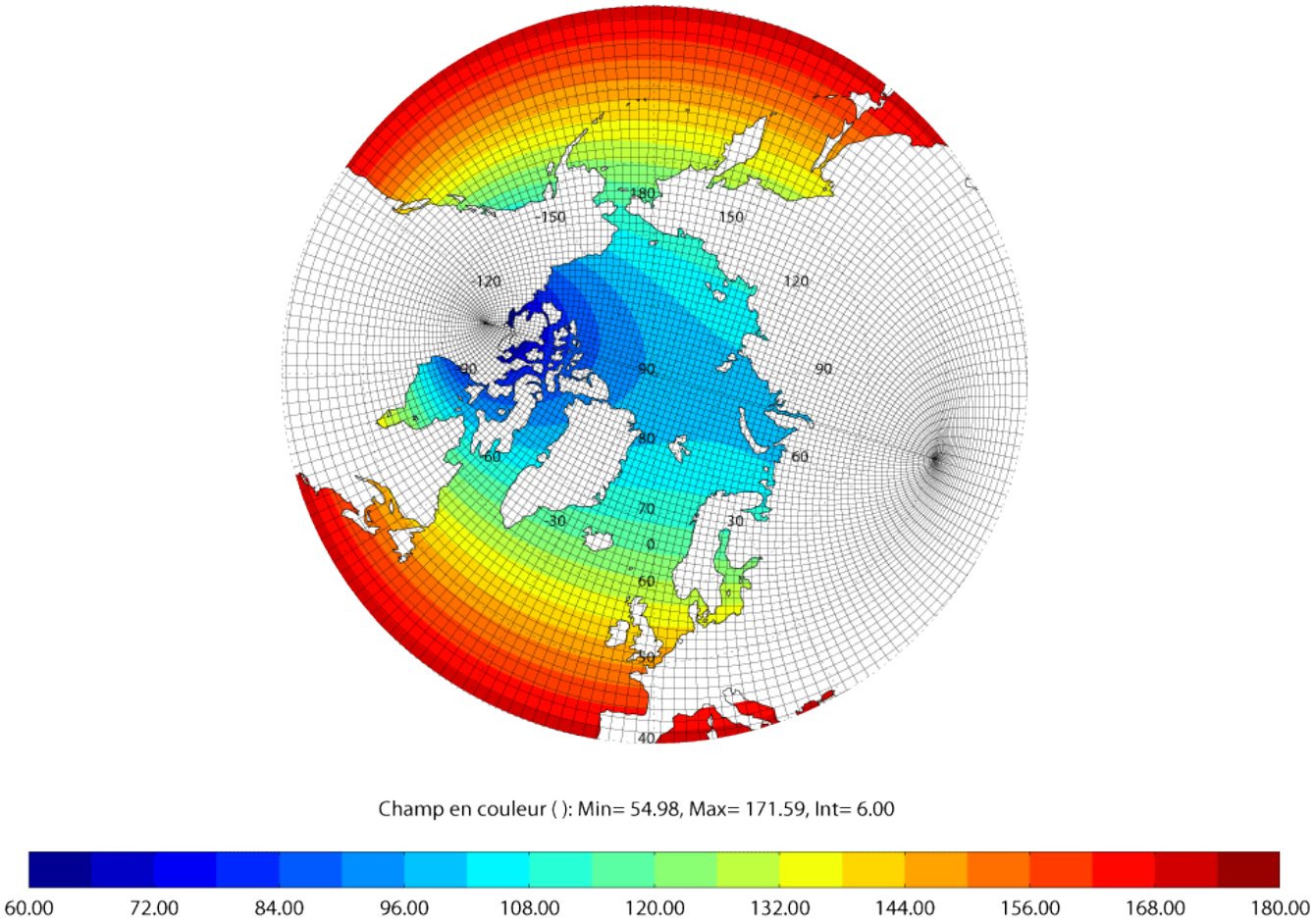
$\mathbf{U} = (u, v, w)$: velocity, T : temperature, p : is the pressure
S : salinity, $D_.$: parametrizations of small scales, ρ : in situ density
ρ_0 : reference density, f : is the Coriolis acceleration
g : gravitational acceleration.

Boundary conditions

- Land-ocean interface: mass exchange of fresh water through river runoff
- Solid earth-ocean interface: no flux of heat and salt across solid boundaries, bottom velocity is parallel to solid boundaries, friction process.
- Atmosphere-ocean interface: continuity of pressure, wind stress, heat exchange, precipitation, evaporation
- Sea ice - ocean interface

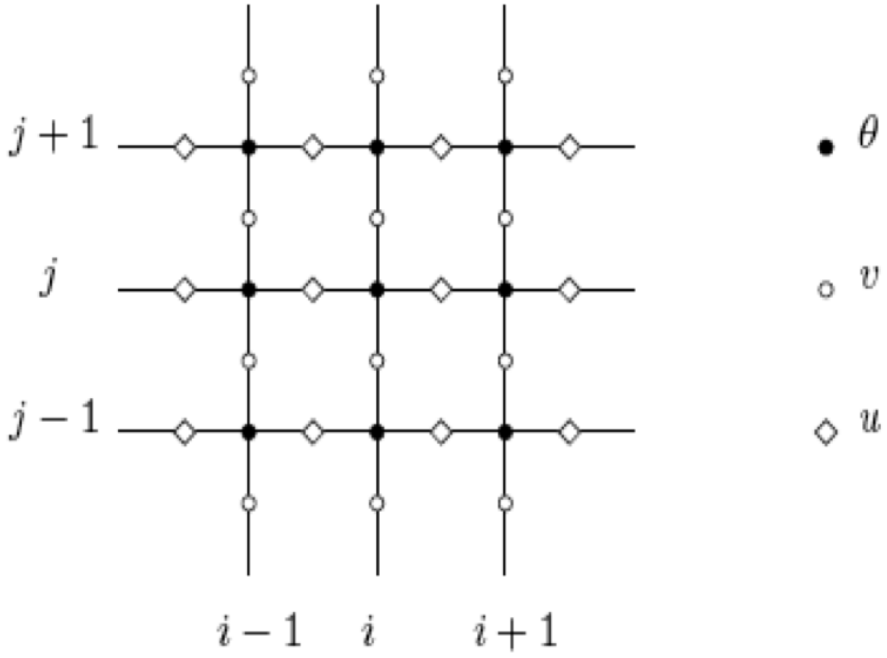
Configuration

- ORCA 2:
  - Global configuration of 2° : $i \times j \times k = 180 \times 149 \times 31$



Discretization

- Space: Finite Difference on Arakawa C grid type



- Time: Leap frog scheme
  - $u^{n+1} = u^{n-1} + 2\Delta t F^n$ non diffusive terms
  - $u^{n+1} = u^{n-1} + 2\Delta t D^{n-1}$ diffusive horizontal terms
  - $u^{n+1} = u^{n-1} + 2\Delta t D^{n+1}$ diffusive vertical terms

Plan

Optimal control theory summary

- Direct model
  The system state equation can be given by

$$\begin{cases} y = \mathcal{M}(x) \\ y(0) = x \end{cases}$$

- Cost function:

$$2\mathcal{J}(x) = 2\mathcal{J}^0(x) + 2\mathcal{J}^b(x) = \|\mathcal{H}(y(x)) - y^0\|^2 + \|x - x^b\|^2$$

  $\mathcal{G}$: observation operator y^0 : observations x^b : background control
- minimization of $\mathcal{J}(x)$ with respect to the control vector x using gradient based algorithm

Incremental 4D-var

- Tangent linear approximation

$$\mathcal{M}(x+\delta x) = \mathcal{M}(x) + M\delta x,\ \delta x = x - x^b$$

$$\mathcal{H}(x+\delta x) = \mathcal{H}(x) + M\delta x,\ \delta x = x - x^b$$

- Cost function:

$$2\mathcal{J}(\delta x) = \|H.M\delta x - d\|^2 + \|\delta x\|^2$$

$d : y^o - \mathcal{H}(\mathcal{M}(x))$

- minimization of $\mathcal{J}(\delta x)$ with respect to the control vector δx using gradient based algorithm
- Quadratic cost function and shorter control vector. Nonlinearities taken by updating x^b.

Tangent and Adjoint Model Development for OPA

- Gradient based algorithm
- 4D-Var:

$$\mathcal{J}(x) \longrightarrow \mathcal{M}$$

$$\nabla\mathcal{J}(x) \longrightarrow M^T$$

- Incremental 4D-Var:

$$\mathcal{J}(\delta x) \longrightarrow M$$

$$\nabla\mathcal{J}(\delta x) \longrightarrow M \text{ and } M^T$$

$\mathcal{M}$: Model OPA
M : Tangent Linear Model of OPA
M^T : Adjoint Model of OPA

NEMO Tangent and Adjoint Model "NEMOTAM" History

- 1992-94: 1st version developed by E. Greiner (LODYC) for OPA4.
  - Applied to 4D-Var with a tropical Atlantic configuration.
- 1995-96: Major rewrite for OPA7 by F. Van den Berghe (CETIIS) and A. Weaver (LODYC).
  - This version was never exploited scientifically.
- 1997-2001: Adapted to OPA8.0 - 8.1 by A. Weaver (LODYC-CERFACS).
  - Developed initially for 4D-Var with a tropical Pacific configuration (TDH).
  - Widely used, primarily for 4D-Var studies.
  - Applied to applications other than data assimilation (singular vectors / optimal perturbations).
- 2002-present: Developed for OPA8.2, free-surface version, by A. Weaver (CERFACS) and C. Deltel (LOCEAN).
  - 1st global ocean version (ORCA2¡).
  - Used for 4D-Var in the ENACT project.
  - Currently used by several groups for a variety of applications.

NEMOTAM Current Status

- Hand written tangent and adjoint model
- OPA 9.0 /NEMO: Major new version in Fortran 95.
- Development for OPA 9.0/NEMO using Automatic Differentiation.

Plan

Automatic Differentiation (AD) of Computer Programs

- Every programming language provides a limited number of elementary mathematical functions
- computer program, no matter how complicated, may be viewed as the composition of these so-called intrinsic functions

$$P = \{I_1; I_2; \ldots; I_{p-1}; I_p\} \text{ implement } F = f_p \circ f_{p-1} \circ \ldots f_1$$

- Derivatives for the intrinsic functions are combined using the chaine rule

$$F'(x_0 = x) = f'_p(x_{p-1}) \cdot f'_{p-1}(x_{p-2}) \cdots f'_1(x_0);\ x_i = f_i(x_{i-1})$$

Automatic Differentiation (AD) of Computer Programs

- But calculating and multiplying jacobians is too expensive

$$F'(x_0 = x) = f'_p(x_{p-1}) \cdot f'_{p-1}(x_{p-2}) \cdots f'_1(x_0)$$

- Tangent mode

$$y = F'(x) \cdot \dot{x} = f'_p(x_{p-1}) \cdot f'_{p-1}(x_{p-2}) \cdots f'_1(x_0)) \cdot \dot{x}$$

- Reverse mode

$$\overline{x} = F'^T(x) \cdot \overline{y} = f'^T_1(x_0) \cdots f'^T_{p-1}(x_{p-2}) \cdot f'^T_p(x_{p-1}) \cdot \overline{y}$$

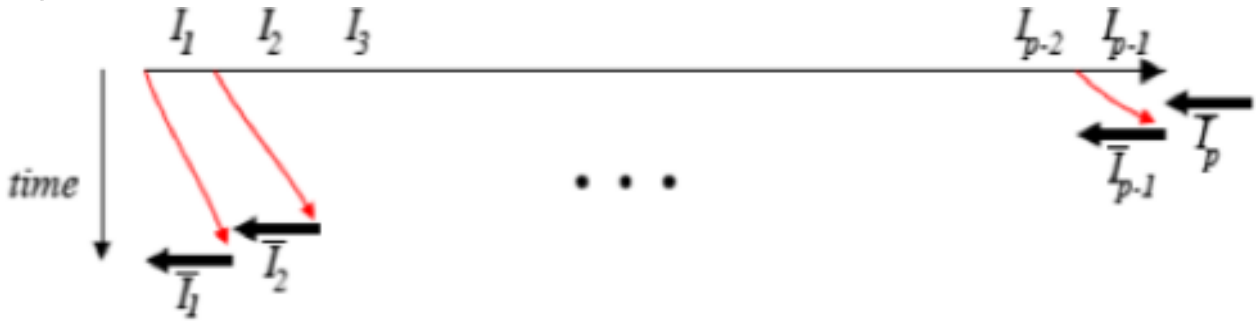
- Which mode ?

$$F : \mathbb{R}^m \longrightarrow \mathbb{R}^n$$

Tangent mode: $m \leq n$
Reverse mode: $m >> n$ (e.g. data assimilation)

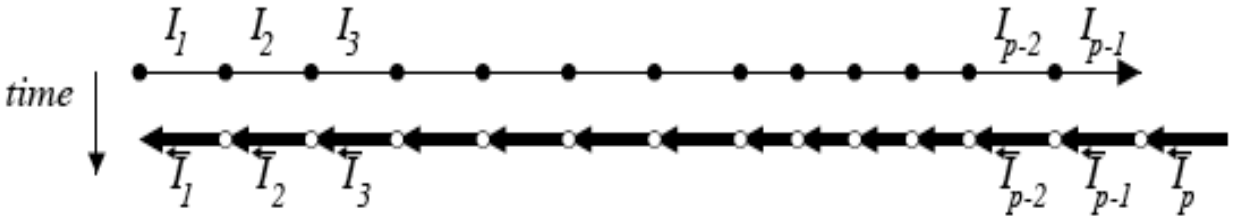
Automatic Differentiation (AD) of Computer Programs : Recompute vs Restore

- Form of adjoint code:

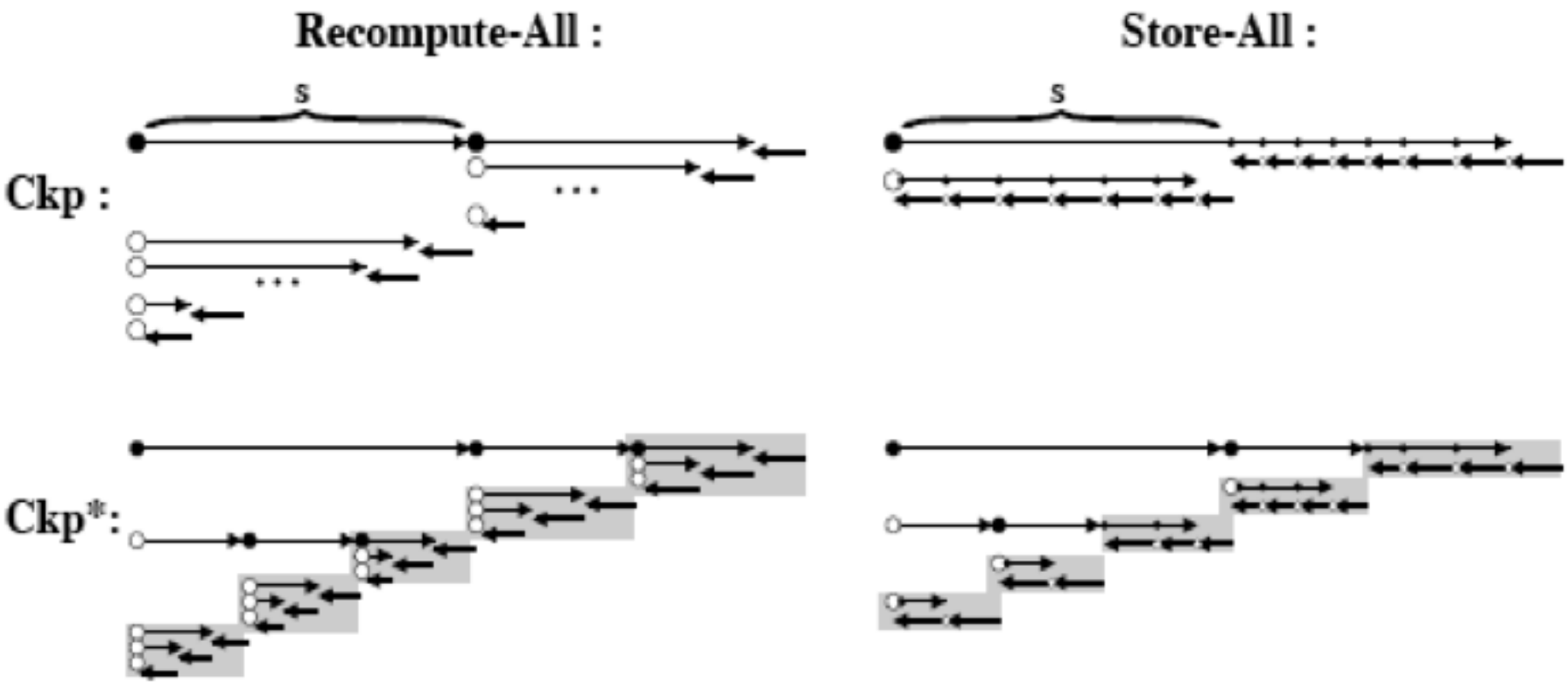


- Recompute all strategy:

- Restore all strategy:



Automatic Differentiation (AD) of Computer Programs : Checkpointing



Validation and Performances

- **Global Configuration:** nitend = 8

DD :	3681856 32812.500
DA Tangent:	3681856 49645.601
DA Inverse (gradient):	3681856 49645.595

Time of original function:	15.0999994874001
Time of tangent AD function:	31.0399990081787
Time of reverse AD function:	69.3400033712387

Max Stack size: 109898 blocks of 16384 bytes => total:1717.156250 Mbytes

Validation and Performances

- **Global Configuration: Zoom on Antarctic** nitend = 35

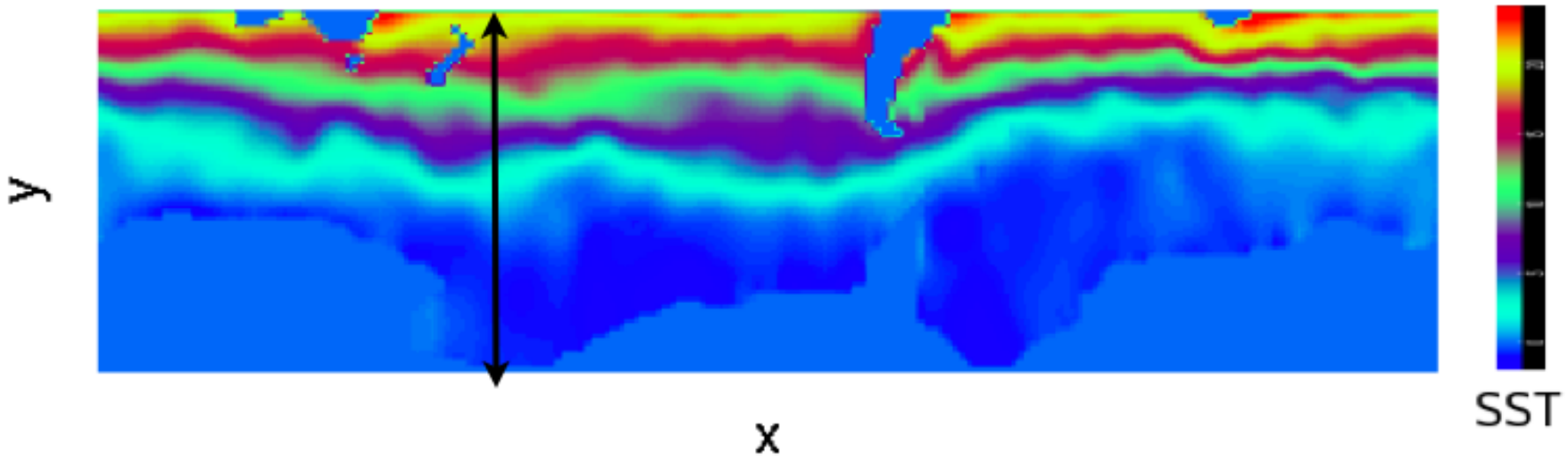
DD :	0.153410689119479663 09E+13
DA Tangent:	0.15341074054565766602E+13
DA Inverse (gradient):	0.15341074054565534668E+13

Time of original function:	22.9399992227554
Time of tangent AD function:	38.5200004577637
Time of reverse AD function:	121.979995727539

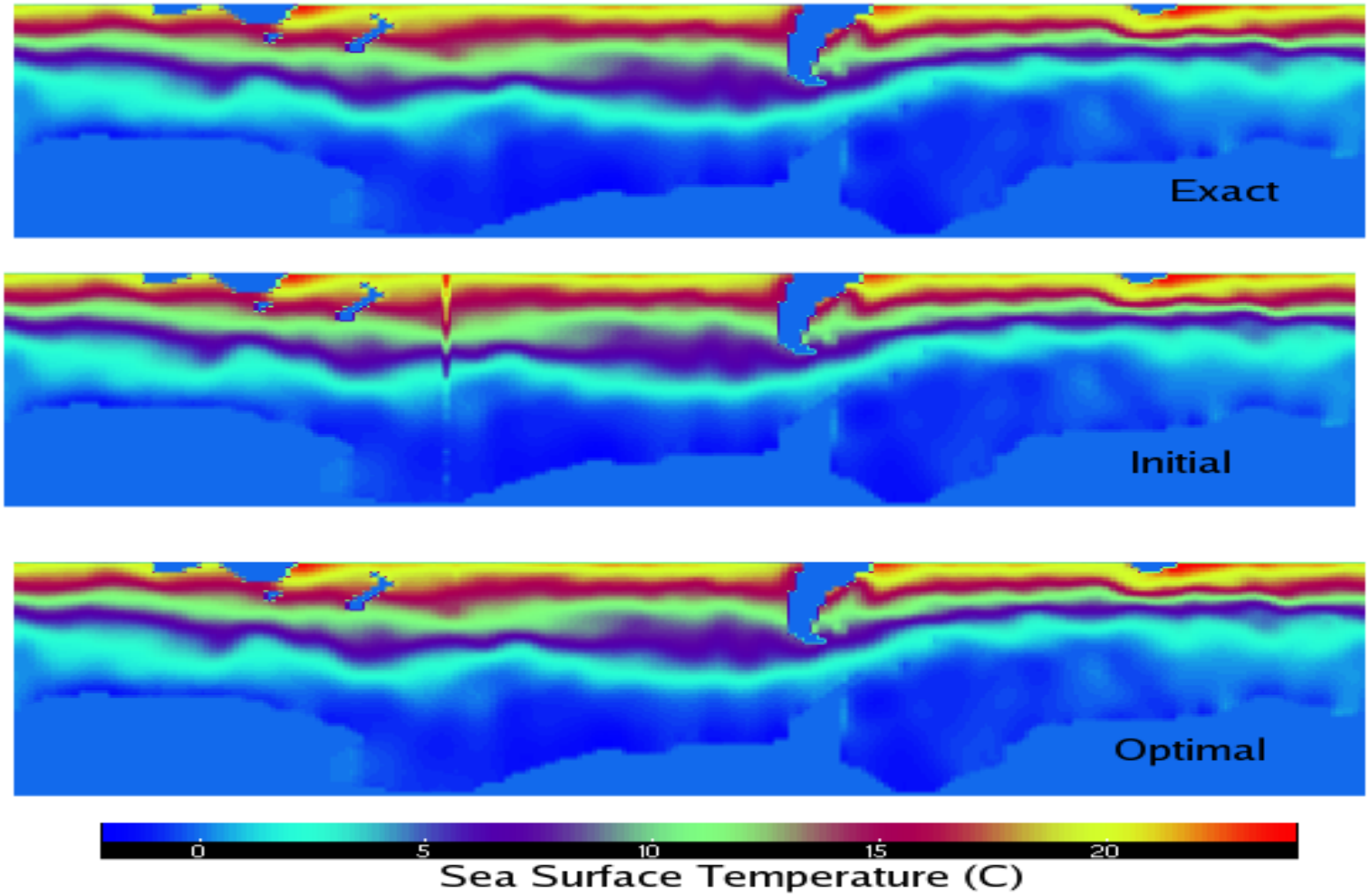
Max Stack size: 153337 blocks of 16384 bytes => total:2395.890625 Mbytes

Twin Experiment

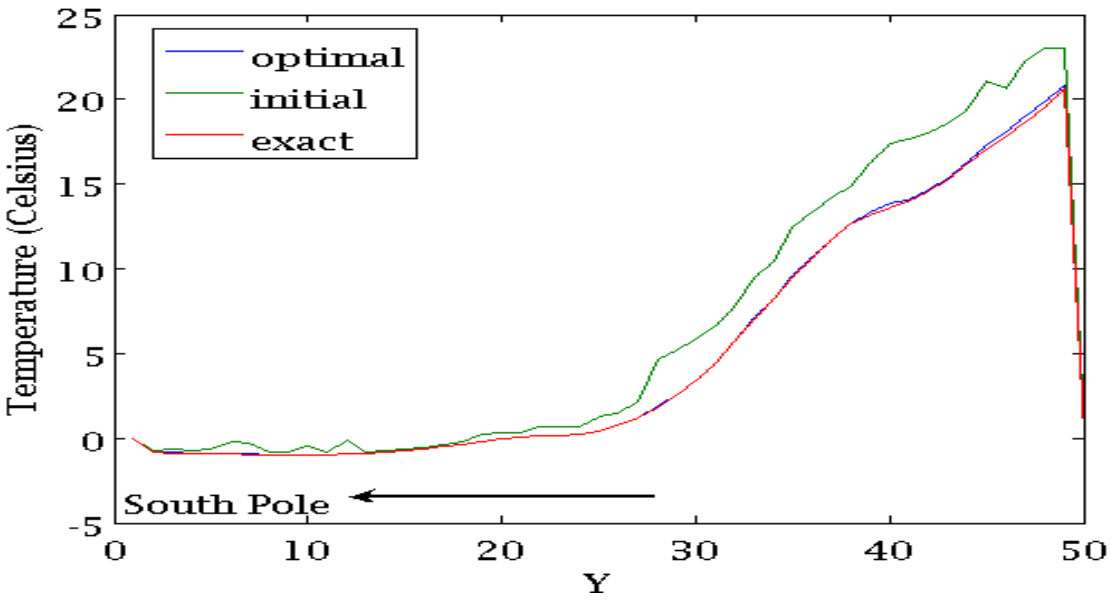
- Zoom on Antarctic
- Fully nonlinear approach
- Distributed observations
- Estimating sea surface temperature at x=60 (longitude)

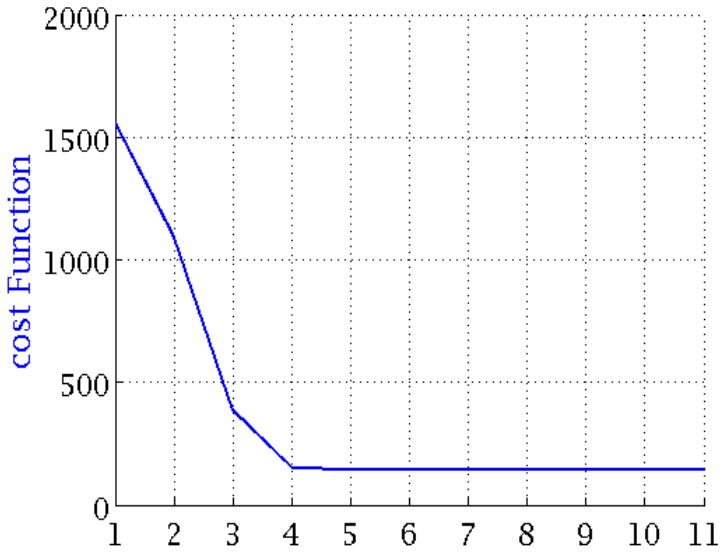


Twin Experiment: Results



Twin Experiments: Results





In Progress

- Optimal checkpointing 'treeverse/revolve'
- Global Configuration
- ...