ADJOINT TECHNIQUE :
GENERAL DESCRIPTION

I. Definitions of tangent linear (TL) and adjoint (AD) operators

$$M : \mathrm{R}^n \to \mathrm{R}^p$$

$$x \to y = M(x) \qquad \text{with } x = (x_i)_{1 \le i \le n} \quad y = (y_j)_{1 \le j \le p}$$

• The tangent linear operator of M in x is the differential application in x : noted $\mathbf{M_x}$

$$\mathbf{M}_x : \mathrm{R}^n \to \mathrm{R}^p$$

$$\delta x \to \delta y = \mathbf{M}_x \cdot \delta x \qquad \text{with } \mathbf{M_x} = \left(\frac{\partial y_j}{\partial x_i}(x)\right)_{1 \le j \le p,\ 1 \le i \le n}$$

$$\forall j \in [1,p], \quad \delta y_j = \sum_{i=1}^{n} \left(\frac{\partial y_j}{\partial x_i}(x)\right) \cdot \delta x_i$$

• Given 2 scalar products of R^n and R^p respectively defined by the matrices S_n and S_p :

$$\begin{cases} \forall\ (x_1, x_2) \in \mathrm{R}^n \times \mathrm{R}^n & \langle x_1 \mid x_2 \rangle = x_1^{\mathrm{T}} \cdot S_n \cdot x_2 \\ \forall\ (y_1, y_2) \in \mathrm{R}^p \times \mathrm{R}^p & \langle y_1 \mid y_2 \rangle = y_1^{\mathrm{T}} \cdot S_p \cdot y_2 \end{cases}$$

Given 2 scalar products S_n and S_p, for any linear application $\mathbf{M} : \mathrm{R}^n \to \mathrm{R}^p$ there is one and only one linear application $\mathbf{M^*} : \mathrm{R}^p \to \mathrm{R}^n$ such as :

$$\forall\ \delta x \in \mathrm{R}^n, \ \forall\ \delta y \in \mathrm{R}^p, \ \langle \mathbf{M} \cdot \delta x \mid \delta y \rangle = \langle \delta x \mid \mathbf{M}^* \cdot \delta y \rangle \qquad (1)$$

$\mathbf{M^*}$ is the adjoint application of the linear application $\mathbf{M}$.

$$(1) \Leftrightarrow (\mathbf{M} \cdot \delta x)^{\mathrm{T}} \cdot S_p \cdot \delta y = (\delta x)^{\mathrm{T}} \cdot S_n \cdot \mathbf{M}^* \cdot \delta y$$

$$\Leftrightarrow (\delta x)^{\mathrm{T}} \cdot \mathbf{M^T} \cdot S_p \cdot \delta y = (\delta x)^{\mathrm{T}} \cdot S_n \cdot \mathbf{M}^* \cdot \delta y$$

$$\Leftrightarrow \mathbf{M^T} \cdot S_p = S_n \cdot \mathbf{M}^*$$

$$\Leftrightarrow \mathbf{M}^* = S_n^{-1} \cdot \mathbf{M^T} \cdot S_p$$

If Euclidian scalar products : $S_n = I_n$ and $S_p = I_p$, $\mathbf{M^* = M^T}$

• Properties

given $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$ such as : $y = M(x)$, with $M : \mathbb{R}^n \to \mathbb{R}^p$ a general application,

given a function $J : \mathbb{R}^p \to \mathbb{R}$,

given $\delta x \in \mathbb{R}^n$ and $\delta y \in \mathbb{R}^p$ such as : $\delta y = \mathbf{M}_x.\delta x$,

$$\delta J = \langle \nabla_x J \mid \delta x \rangle = \langle \nabla_y J \mid \delta y \rangle = \langle \nabla_y J \mid \mathbf{M_x} \cdot \delta x \rangle = \langle \mathbf{M_x^*} \cdot \nabla_y J \mid \delta x \rangle$$

$$\nabla_x J = \mathbf{M_x^*} \cdot \nabla_y J$$

Other demonstration with the Euclidian scalar products :

$$\left(\frac{\partial J}{\partial x_i}\right)_{1 \le i \le n} = \left(\frac{\partial y_j}{\partial x_i}(x)\right)_{1 \le j \le p,\ 1 \le i \le n} \cdot \left(\frac{\partial J}{\partial y_j}\right)_{1 \le j \le p} \ : \ \nabla_x J = \mathbf{M_x^*} \cdot \nabla_y J = \mathbf{M_x^T} \cdot \nabla_y J$$

II. Case of 3D-Var and 4D-Var assimilation

3D-Var

$$J(x) = \underbrace{\tfrac{1}{2}(x - x_B)^{\mathrm{T}} \cdot \mathbf{B}^{-1} \cdot (x - x_B)}_{J_B} + \underbrace{\tfrac{1}{2}(H(x) - y)^{\mathrm{T}} \cdot \mathbf{R}^{-1} \cdot (H(x) - y)}_{J_O}$$

$$\nabla_x J = \nabla_x J_B + \nabla_x J_O$$

with :

$$\nabla_x J_B = \mathbf{B}^{-1} \cdot (x - x_B) \qquad \nabla_x J_O = \mathbf{H^T} \cdot \nabla_H J_O = \mathbf{H^T} \cdot \mathbf{R^{-1}} \cdot (H(x) - y)$$

$$\nabla_x J = \mathbf{B^{-1}} \cdot (x - x_B) + \mathbf{H^T} \cdot (H(x) - y)$$

4D-Var

$$J_O(x) = \sum_{i=1}^{N} \tfrac{1}{2}\Big(H_i\big(M_i(x)\big) - y_i\Big)^{\mathrm{T}} \cdot \mathbf{R}_i^{\mathbf{-1}} \cdot \Big(H_i\big(M_i(x)\big) - y_i\Big)$$

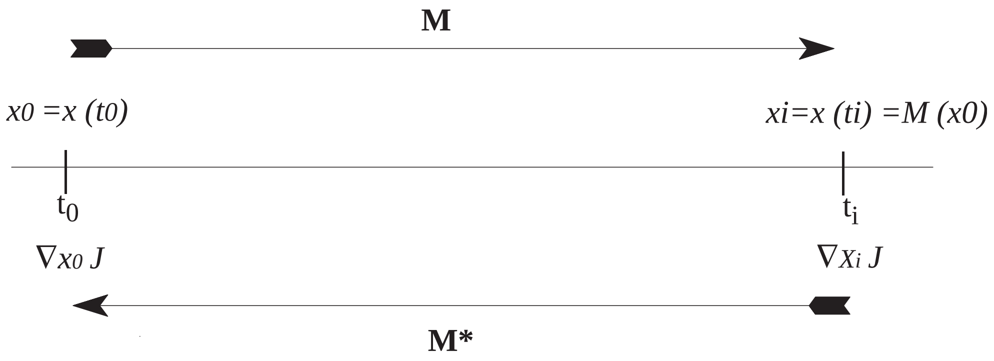
$$\nabla_x J_O = \sum_{i=1}^{N} \mathbf{M}_i^{\mathbf{T}} \cdot \mathbf{H}_i^{\mathbf{T}} \cdot \mathbf{R}_i^{\mathbf{-1}} \cdot \Big(H_i\big(M_i(x)\big) - y_i\Big)$$

Properties

Talagrand 1988 :

The direct forecast model is needed in 4D-Var to carry forward in time the value X of the model from the initial time t_0 of the assimilation period to any observation time t_i.

The adjoint model is needed in 4D-Var to carry backward in time the the gradient of a cost function from any time t_i to the initial time t_0 of the assimilation period.



Example

Given ξ : space of functions $u(x,t)$ periodic in x over the interval [0,L] and a scalar product :

$$\langle u_1 \mid u_2 \rangle = \int_0^L u_1 u_2 dx$$

$$\Rightarrow \langle u_1 \mid \frac{\partial u_2}{\partial x} \rangle = \int_0^L u_1 \frac{\partial u_2}{\partial x} dx = \left[u_1 u_2 \right]_0^L - \int_0^L \frac{\partial u_1}{\partial x} u_2 dx = -\langle \frac{\partial u_1}{\partial x} \mid u_2 \rangle$$

The adjoint of the operator : $\delta u \rightarrow \partial(\delta u) / \partial x$ is : $\delta v \rightarrow -\partial(\delta v) / \partial x$

$$\Rightarrow \langle u_1 \mid \frac{\partial^2 u_2}{\partial x^2} \rangle = \langle \frac{\partial^2 u_1}{\partial x^2} \mid u_2 \rangle$$

The adjoint of the operator $\partial^2 / \partial x^2$ is $\partial^2 / \partial x^2$! Self-adjoint operator

Ex : The Laplacian is a self-adjoint operator

<u>III. Practical rules for coding TL and AD</u>

By hand : fastidious, precise work $\rightarrow$ readable and efficient code

Automatic code generators (ex: TAMC)

TL writing

- not very difficult
- derivation line by line

- storage or recomputation of the trajectory
- `MIN, MAX, ABS,` ... $\rightarrow$ `IF, THEN, ELSE`

Direct code

```
.
ZA=EXP(ALPHA+BETA*XA**2)
ZB=ZA*MAX(0.,XA)
.
.
.
.
.
.
.
.
```

TL code

```
 .
 ZA=EXP(ALPHA+BETA*XA5**2)
&   *2*BETA*XA5*XA
 ZA5=EXP(ALPHA+BETA*XA5**2)
 IF (XA5 >= 0.) THEN
   ZB=ZA5*XA+XA5*ZA
 ELSE
   ZB=0.
 ENDIF
 ZB5=ZA5*MAX(0.,XA5)
 .
```

AD writing

$$\mathbf{M = M_n\ M_{n-1} \dots M_2\ M_1} \quad \rightarrow \quad \mathbf{M^T = M_1^T\ M_2^T \dots M_{n-1}^T\ M_n^T}$$

- computing the trajectory (be careful)
- setting the local variables to zero
- adjoint coding from the end to the beginning

TL code

```
ZY=A1*ZX1+...+AN*ZXN+B*ZY
```

AD code

```
ZX1=ZX1+A1*ZY
.
.
ZXN=ZXN+AN*ZY
ZY=B*ZY
```

$$\begin{vmatrix} ZX1 \\ ZX2 \\ . \\ . \\ ZXN \\ ZY \end{vmatrix} = \begin{vmatrix} 1 & 0 & . & . & 0 \\ 0 & 1 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & . & 1 & 0 \\ A1 & A2 & . & AN & B \end{vmatrix} \times \begin{vmatrix} ZX1 \\ ZX2 \\ . \\ . \\ ZXN \\ ZY \end{vmatrix} \qquad \begin{vmatrix} ZX1 \\ ZX2 \\ . \\ . \\ ZXN \\ ZY \end{vmatrix} = \begin{vmatrix} 1 & 0 & . & . & A1 \\ 0 & 1 & . & . & A2 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & . & 1 & AN \\ 0 & 0 & . & 0 & B \end{vmatrix} \times \begin{vmatrix} ZX1 \\ ZX2 \\ . \\ . \\ ZXN \\ ZY \end{vmatrix}$$

Validation

- Tangent linear model

$$\frac{M(x+\lambda\cdot\delta x)-M(x)}{M'(\lambda\cdot\delta x)} \xrightarrow[\lambda\to 0]{} 1$$

$\lambda = 10^{-1}$	$r =$ **0.9**478065803
$\lambda = 10^{-2}$	$r =$ **0.99**67156712
$\lambda = 10^{-3}$	$r =$ **0.999**3278142
$\lambda = 10^{-4}$	$r =$ **0.9999**124930
$\lambda = 10^{-5}$	$r =$ **0.99999**35082
$\lambda = 10^{-6}$	$r =$ **0.999999**0391
$\lambda = 10^{-7}$	$r =$ **0.99999**73140
$\lambda = 10^{-8}$	$r =$ **0.9999**214307
$\lambda = 10^{-9}$	$r =$ **0.999**0174236
$\lambda = 10^{-10}$	$r =$ **0.99**23902124

- Adjoint model

$$\forall\, x,\ \forall\, y,\ \ \langle Mx \mid y\rangle = \langle x \mid M^{*} y\rangle$$

$\langle Mx \mid y\rangle = -$0.13765102625251640000

$\langle x \mid M^{*}y\rangle = -$0.13765102625251680000

`The difference is 11.551 times the precision of the machine`

IV. Different applications of AD models, validation

- Variational data assimilation ⟵ 3D-Var, 4D-Var
- Singular vector computation ⟵ Initial perturbations for EPS
- Sensitivity analysis ⟵ Forecast errors
- Efficient determination of model parameters