

# CAPACITES ACTUELLES DE LA DIFFERENTIATION AUTOMATIQUE: L'ADJOINT D'OPA 9.0 PAR TAPENADE

Bruno Ferron, Laurent Hascoët

Bruno.Ferron@ifremer.fr  
Laurent.Hascoet@sophia.inria.fr

Colloque Assimilation de Données, Toulouse, Mai 2006

- 1 Très brève présentation de la DA
- 2 Avancement actuel de la DA
- 3 TAPENADE 2.1
- 4 TAPENADE sur OPA 9.0/GYRE

# Différentier des Programmes !?

Programme = composition de fonctions, différentiable  
(... par morceaux si contrôle.)

$P : \{l_1; l_2; \dots; l_{p-1}; l_p; \}$  implémente  $F = f_p \circ f_{p-1} \circ \dots \circ f_1$   
donc  $F'(x_0 = x) = f'_p(x_{p-1}) \cdot f'_{p-1}(x_{p-2}) \cdot \dots \cdot f'_1(x_0)$

D'où un programme différentié  $P'$ ,

(re)construit **automatiquement** depuis  $P$ ,  
qui calcule des dérivées **analytiques**.

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

... mais calculer et multiplier des Jacobiennes est cher !

⇒ Différentiation **tangente** :

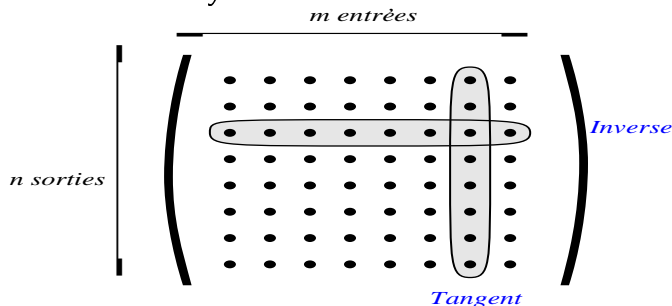
$$\dot{y} = F'(x) \cdot \dot{x} = f'_p(x_{p-1}) \cdot f'_{p-1}(x_{p-2}) \cdot \dots \cdot f'_1(x_0) \cdot \dot{x}$$

⇒ Différentiation **inverse** :

$$\bar{x} = F'^t(x) \cdot \bar{y} = f_1'^t(x_0) \cdot \dots \cdot f_{p-1}'^t(x_{p-2}) \cdot f_p'^t(x_{p-1}) \cdot \bar{y}$$

# Quel mode pour quelles dérivées ?

$$F : x \in \mathbb{R}^m \mapsto y \in \mathbb{R}^n$$



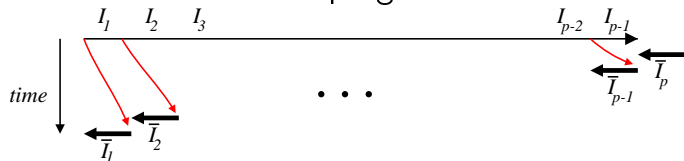
- $F'(x)$  coûte  $m * 4 * P$  en mode tangent  
bon si  $m \leq n$
- $F'(x)$  coûte  $n * 4 * P$  en mode inverse  
bon si  $m \gg n$  (e.g.  $n = 1$  en assimilation !)

# Modèle de Différentiation Inverse

Par construction:

$$\bar{x} = F'^t(x) \cdot \bar{y} = f_1'^t(x_0) \cdot \dots \cdot f_{p-1}'^t(x_{p-2}) \cdot f_p'^t(x_{p-1}) \cdot \bar{y}$$

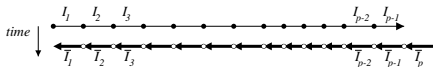
qui détermine la forme d'un programme inverse:



Comment organiser la restauration inverse des valeurs intermédiaires?

# Restauration en ordre inverse

Tout stocker



Cher en mémoire

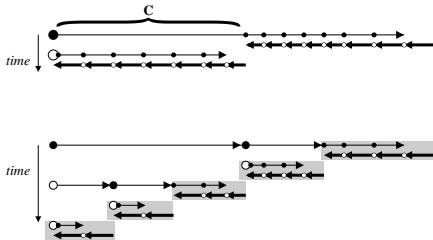
Tout recalculer



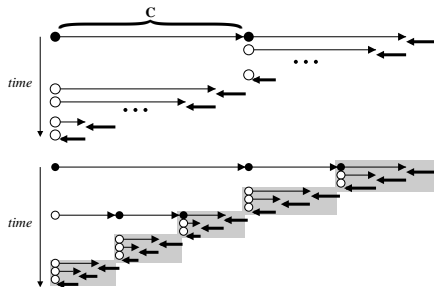
Cher en temps

# Compromis stockage/recalcul ; Checkpointing

## Tout stocker



## Tout recalculer



finalement très semblables !



- 1 Très brève présentation de la DA
- 2 **Avancement actuel de la DA**
- 3 TAPENADE 2.1
- 4 TAPENADE sur OPA 9.0/GYRE

# Les modes de différentiation




Tangent  
Tangent\*

Inverse  
Inverse\*

Tangent-sur-Inverse  
 $v^t * H * v$

Taylor directionnel

Jacobienne étendue  
Series Laurent



# Raffinements du modèle de la DA



Activité+Utilité

Ordre calculs dérivées (NP)

Inversion du contrôle

Détection stockages inutiles

Détection calculs inutiles


Recalculs grain fin

Snapshots optimaux

Schémas optimaux de Checkpointing



# Langages et constructions reconnus



C++

malloc  
Objets

pointeurs


MPI

C Matlab

OpenMP

Fortran 95

Fortran 77



cf [www.autodiff.org](http://www.autodiff.org)

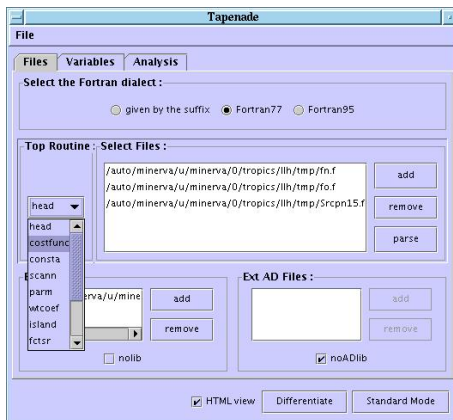
- **Adol-C** (Surcharge):  
tangent, inverse, dérivées supérieures.
- **NagWare F95** (Compilateur, surcharge):  
tangent, inverse
- **Adi(j)for, AdiC, OpenAD** (Transfo.):  
tangent, inverse, "Tout stocker" + Checkpointing
- **TAMC, TAF, TAC++** (Transfo.):  
tangent, inverse, "tout recalculer" + Checkpointing
- **Tapenade** (Transfo.):  
tangent, inverse, "Tout stocker" + Checkpointing

# Exemples d'applications récents (mode inverse)

- **Adol-C**: contrôle du bras d'un robot
- **Adifor**: CFD (codes SEPRAN, TFS)
- **OpenAD**: MIT gcm simplifié (ECCO-2 01/06)
- **TAMC**: MIT ocean gcm
- **Tapenade**: CFD (bang sonique, optimisation de maillages)

- 1 Très brève présentation de la DA
- 2 Avancement actuel de la DA
- 3 TAPENADE 2.1
- 4 TAPENADE sur OPA 9.0/GYRE

# Interface d'appel (web ou local)



ou bien ...

```
$> tapenade -b -msglevel 30 -head psiroe -vars  
"x y t" -outvars "ce" psiroe.f lib.f
```



# Résultat (web ou local)

**Original call graph**

- adj
  - sub2
  - sub1
  - maxx

**Differentiated call graph**

- adj\_dv
  - maxx\_dv
  - sub1\_dv
  - sub2\_dv

```

SUBROUTINE ADJ(u, z, t)
  REAL t, u, z
  REAL x(14), y
  COMMON /cc/ x, y
  INTEGER i, MAXX
  REAL v
  EXTERNAL MAXX

  i = 5
  x(1) = y * u + t
  z = MAXX(z, t)
  u = 0.0
  CALL SUB1(u, x(i), z, v)
  t = t + x(1) * z + 3 * v
  y = 0.0
  i = 6
  CALL SUB2(u, x(3), z, v)

```

```

x(1) = y * u + t
CALL MAXX_DV(z, zd, t, td, z)
u = 0.0
CALL SUB1_DV(u, ud, x(1), xd(1))
DO nd=1,nbndirs
  td(nd) = td(nd) + z * xd(nd)
ENDDO
t = t + x(1) * z + 3 * v
y = 0.0
i = 6
CALL SUB2_DV(u, ud, x(3), xd(1))
DO nd=1,nbndirs
  td(nd) = td(nd) + z * xd(nd)
ENDDO
t = t + x(1) * z + 3 * u
DO nd=1,nbndirs
  zd(nd) = 0.0
ENDDO

```

2 adj: undeclared external routine: maxx  
3 adj: Return type of maxx set by implicit rule to INTEGER  
4 adj: argument type mismatch in call of sub1, REAL(0:6) expected, receives I  
5 adj: argument type mismatch in call of sub2, REAL(0:12) expected, receives I  
6 maxx: Tool: Please provide a differentiated function for unit maxx for argu

<b>Programme d'origine</b>	<b>Tapenade inverse: 1/2</b>
<pre>x = x**3 CALL SUB(a, x, 1.5, z) x = x*y</pre>	<pre>CALL PUSHREAL4(x) x = x**3 CALL PUSHREAL4(x) CALL SUB(a, x, 1.5, z) x = x*y</pre>
<b>Tapenade tangent</b>	<b>Tapenade inverse: 2/2</b>
<pre>xd = 3*x**2*xd x = x**3 CALL SUB_D(a, ad, x, xd,            1.5, 0.0, z) xd = y*xd x = x*y</pre>	<pre>xb = y*xb CALL POPREAL4(x) CALL POPREAL4(x) CALL SUB_B(a, ab, x, xb,            1.5, arg2b, z) CALL POPREAL4(x) xb = 3*x**2*xb</pre>

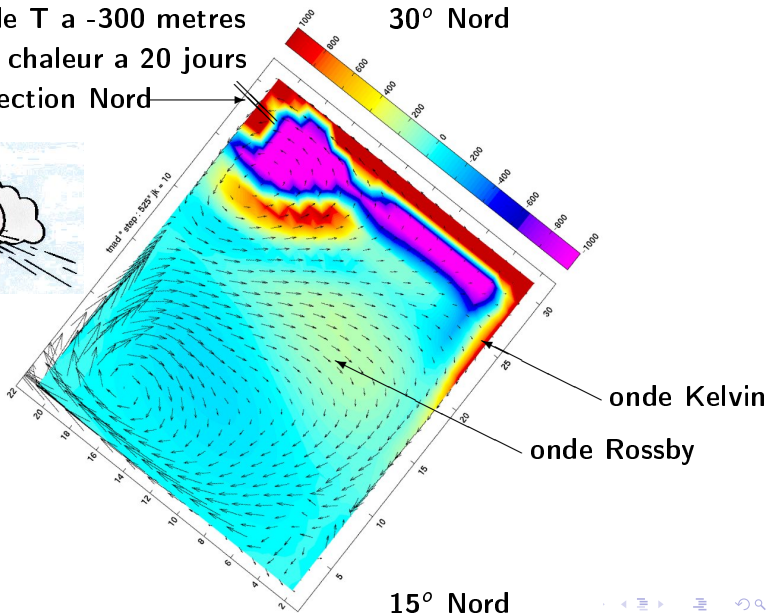
- **Portail DA:** <http://www.autodiff.org>
- **Serveur Tapenade:**  
<http://tapenade.inria.fr:8080/tapenade>
- **ftp Tapenade:**  
<ftp://ftp-sop.inria.fr/tropics/tapenade>
- **Documentation:**  
<http://www-sop.inria.fr/tropics>  
<http://www.inria.fr/rrrt/rt-0300.html>

- 1 Très brève présentation de la DA
- 2 Avancement actuel de la DA
- 3 TAPENADE 2.1
- 4 TAPENADE sur OPA 9.0/GYRE

- **OPA:** (LOCEAN/LODYC, Paris VI)  
Modèle de circulation océanique générale.
- **OPA 9.0:** Fortran95, 40 000 lignes
- **GYRE:** (M. Levy)  
configuration bassin rectangulaire fermé + vent  
 $32 \times 22 \times 31$  mailles, 4320 pas de temps (20 jours)

# Gradient TAPENADE

Influence de T a -300 metres  
sur flux de chaleur a 20 jours  
a travers section Nord



onde Kelvin

onde Rossby

15° Nord

- **Validation:**

DD : 366.77744614799468081

DA Tangent: 366.56522267259947512

DA Inverse (gradient): 366.56522267259947512

- **Performances:**

Temps de différentiation négligeable!

Simulation: 92 s 86 Mo

Gradient: 657 s 580 Mo

×7.1 (→ 5.5) ×6.7 (→ 28)

- Configuration ORCA-2
- Checkpointing optimal "treeverse/revolve"
- Adjoint "adapté"