14th European Workshop on AD

ADJOINTS OF FIXED-POINT ITERATIONS

Ala Taftaf

INRIA, SOPHIA ANTIPOLIS, FRANCE

Adjoint Algorithms

- Adjoint algorithms : the most efficient way to obtain gradient of a numerical simulation

- Computing the gradients has a cost (time, memory)

- One way around : take advantage of the structures of the given program (parallel loops, fixed point methods,..)

Fixed Point Iterations

- Many implicit functions $F(z, x) = 0$ are computed with convergent iterations:

  Initial guess $z_0(x)$

  Iterate $z_{k+1}(x) = \varphi(z_k(x), x)$

  To fixed point $z_*(x) = \varphi(z_*(x), x)$

- Followed by an objective function $y = f(z_*, x)$

Adjoint of Fixed Point Iterations

- First iterations of a fixed-point search :
  $z_0(x)$
  $z_1(x) = \varphi(z_0(x), x)$
  .....
  operate on a meaningless state vector =>No adjoint needed
- Adjoin only the (few) last iterations
- Save trajectory storage

At least two authors have studied mathematically fixed point iterations with the goal of defining an efficient adjoint : Christianson and Griewank

Christianson's strategy (BC)

$$\boxed{z_{k+1} = \varphi(z_k(x), x)}$$

z_*

Forward Sweep

$$y = f(z_*, x)$$

$$\bar{y} = 1$$

$$\bar{z} = \bar{y}.f_z(z_*, x)$$

$$\bar{x} = \bar{y}.f_x(z_*, x)$$

$$\bar{w}_0 = \bar{z}$$

$$\boxed{\bar{w}_{k+1} = \bar{w}_k.\varphi_z(z_*, x) + \bar{z}}$$

$\bar{w}_*$

Backward Sweep

$$\bar{x} = \bar{w}_*.\varphi_x(z_*, x) + \bar{x}$$

Griewank's strategy (AG) : Piggyback

$$w_k = F(z_k, x)$$

$$z_{k+1} = z_k - P_k . w_k$$

z_*

$$y = f(z_*, x)$$

Original Program

$$w_k = F(z_k, x)$$

$$\bar{z}_k = \bar{w}_k . F_z(z_k, x) + \bar{y} . f_z(z_k, x)$$

$$z_{k+1} = z_k - P_k . w_k$$

$$\bar{w}_{k+1} = \bar{w}_k - \bar{z}_k . P_k$$

$z_*,\ \bar{w}_*$

$$\bar{x}_* = \bar{w}_* . F_x(z_*, x) + \bar{y} . f_x(z_*, x)$$

Adjoint Program

AG and BC: Similarities

- Both manage to avoid naïve inversion of the original sequence of iterations => Save trajectory storage
- Stopping criterion of the adjoint fixed point is distinct from the original test
- Convergence rate is similar for the derivative computation and the original computation

AG and BC: Differences

	AG	BC
Shape of iteration step	Additional Assumptions $z_{k+1} = z_k - P_k . w_k$	General $z_{k+1} = \varphi(z_k, x)$
Start of adjoining	From the Beginning (more iterations)	After a total convergence
Sequel of F.P. loop	Adjoin repeatedly	Adjoin once

Reasons for our choice

Mostly for implementation reasons, we want :

- **No assumption for iteration shape**
- **no multiple differentiation of the sequel**
- **differentiate only the (few) last iterations**
- **to preserve the 2 sweeps structure of the adjoint code**

Identifying the F.P. structure

In the computation of the adjoint ($\bar{w}_{k+1} = \bar{w}_k . \varphi_z(z_*, x) + \bar{z}$) we need to determine (φ, z, x) in the original program => Adding new directives

z x

```
$AD Begin FP loop
REPEAT UNTIL ( z stationary)
        $AD Begin FP iteration
        z = φ(z, x)
        $AD END FP iteration
END REPEAT
$AD END FP loop
y = f(z, x)
```

Discussion

- Is a differentiation strategy for the iterative computations really needed in practice, as yet more focused strategies exist ?
- Have you samples of code where this strategy could be effective ?

Discussion

- Is a differentiation strategy for the iterative computations really needed in practice, as yet more focused strategies exist ?

- Have you samples of code where this strategy could be effective ?

Thank you for your attention!