Differentiation using multi-complex numbers

Daulet Moldabayev, Heruy Mengistu, Alessandro Arrigoni.
supervisor Thierry Dargent (Thales Alenia Space)

MathMods Master Course
University of l'Aquila
University of Nice

July 16, 2010

- **Motivation** : The precise calculation of derivatives is very essential in many fields, i.e. optimization design, stability and control problems, aerospace engineering.
  As solutions for that there have been developed some several strategies which have to satisfy the following
- **Criteria**
- *accuracy*
- *computational expense*

If is required the *implementation* before use the strategy, of course its easiness will play a big role as well.

Those methods could be classified in the following way, considering their accuracy :

- **Analytical methods** :
  - Automatic Differentiation (AD) → accurate to machine working precision.

    http://www.autodiff.org/

  - Computation by hands.
- **Numerical methods** : i.e.
  - Finite difference (FD)→ central scheme, forward scheme...
  - Complex and Multicomplex method.

The difference is that the first one is going to find an *expression* of the derivative, the second one just its *value* in one point.

Objectives

- Develop a library of programs which allows to work with multicomplex numbers.
- Perform a sensitivity analysis on multicomplex-step method, compaing it with other methods.
- Apply the method for an optimization problem.

A multicomplex number is a natural generalization of a complex number, with different possible definition. We chose the following one:

$$\mathbb{C}^n := \{z_1 + i_n z_2 \mid z_1, z_2 \in \mathbb{C}^{n-1}\}, \quad i_n^2 = -1.$$

where $\mathbb{C}^1 = \mathbb{C}$ *and* $\mathbb{C}^0 = \mathbb{R}$

The benefit of this definition is that it makes implementation very easy: all the number types and functions may be defined recursively.

- Generally we can represent any multicomplex number of order n, $\mathbb{C}^n$, by 2^n coefficients in $\mathbb{R}$: one coefficient, x_0, *for real part*, $x_1 \ldots x_n$coefficients for pure imaginary directions.

$$\mathbb{C}^n = x_0 + x_1 i_1 + x_2 i_2 + \ldots + x_n i_n + x_{12} i_1 i_2 + \ldots + x_{n-1n} i_{n-1} i_n + \ldots + x_{1\ldots n} i_{1\ldots n} \quad (1)$$

where $x_0, \ldots, x_n, \ldots, x_{1\ldots n} \in \mathbb{R}$

The trick is use the fact that a holomorphic function admits a Taylor series expansion around a point z_0.

$$f(x+hi_1+hi_2+..+hi_n) = f(x)+(i_1+..+i_n)\cdot h\cdot f'(x)+(i_1+..+i_n)^2\cdot h^2\cdot\frac{f''(x)}{2}+..+\mathcal{O}(h^3)$$

We used multinomial theorem

$$(i_1+i_2+..+i_n)^k = \sum_{k_1\dots k_n} \frac{n!}{k_1!..k_n!} \times (i_1^{k_1}\dots i_n^{k_n})$$

where $k_1+..+k_n = k$, and we got

$$f^{(n)}(x) = \frac{Im_{1..n}(f(x+hi_1+hi_2+...+hi_n))}{h^n} + \mathcal{O}(h^2) \qquad (2)$$

Here some application for some derivatives:

$$\frac{\partial^2 f(x,y)}{\partial x^2} \approx \frac{Im_{12} f(x + hi_1 + hi_2, y))}{h^2} \tag{3}$$

$$\frac{\partial^2 f(x,y)}{\partial y^2} \approx \frac{Im_{12} f(x, y + hi_1 + hi_2))}{h^2} \tag{4}$$

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{Im_1 f(x + hi, y + hi_2)}{h} \tag{5}$$

Also is possible the computation of partial derivatives.

Declaration of Multicomplex numbers in Fortran

- Bicomplex numer

```
TYPE bicx
  double complex :: a, b
end TYPE bicx
```

- Tricomplex numer

```
TYPE tricx
  TYPE ( bicx ) :: a, b
end TYPE tricx
```

- Quadracomplex

```
TYPE qcx
  TYPE ( tricx ) :: a, b
END TYPE qcx
```

Arithmetic for multicomplex numbers

- To add two bicomplex number,

```
Function add( q1 , q2 ) RESULT ( q3 )
TYPE ( bicx ), intent ( in ) :: q1, q2
TYPE ( bicx ) :: q3
  q3 % a = q1 % a + q2 % a
  q3% b = q1%b + q2% b
End function add
```

- To multiply twoTricomplex number,

```
Function mult( q1, q2 ) RESULT ( q3 )
TYPE ( tricx ), intent ( in ) :: q1, q2
TYPE ( tricx ) :: q3
  q3 % a = q1 % a * q2 % a - q1 % b * q2 % b
  q3 % b = q1 % a * q2 % b + q1 % b * q2 % a
end Function mult
```

Over loading of operators

- For Addition,
  Interface operator (+)
  Module Procedure
  $add - 2cx - 2cx, add - 2cx - r, add - r - 2cx, add - 3cx - 3cx$
  end Interface operator (+)

- equal
  Interface operator (=)
  module procedure equal-2cx-2cx, equal-3cx-3cx
  end Interface operator (=)

- Similarly other operations and higher multicomplex number can be defined.

- Comparison between Multicomplex-step differentiation and finite deference schemes by taking the first ,second and third order derivatives of the following function.

$$f(x) = \frac{\exp(x)}{\sin(x^3) + \cos(x^3)} \tag{6}$$

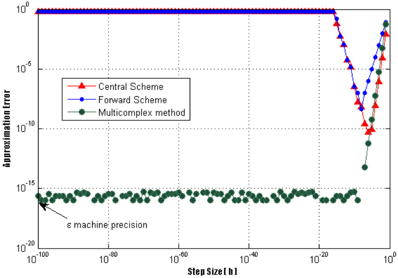

Figure: Approximation errors in 1st order derivative computation.

- Second order derivatives

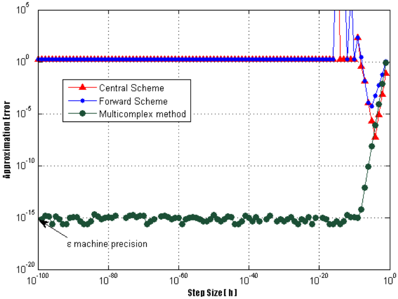

Figure: Approximation errors in 2nd order derivative computation.

- Third order derivatives

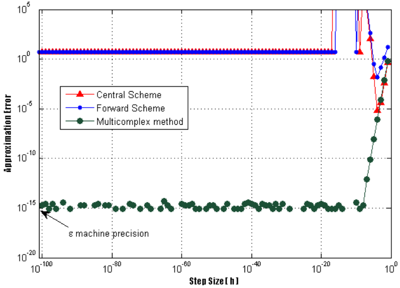

Figure: Approximation errors in 3rd order derivative computation.

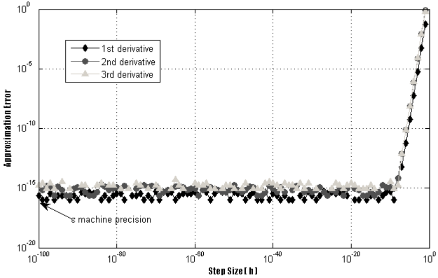

Figure: 1st, 2nd and 3rd order derivatives computed with Multicomplex method.

Relative Computational Time for h = 1.0E-003.

Method	TIME(IN sec)
Analytical	1.00
Multicomplex	1.18029286
Forward (FD)	1.40329469
Central (FD)	1.15070165

Let us apply the Multicomplex method for solution of the following optimization problem:

$$f(x,y) = e^{(x^2-2y)} \quad \longrightarrow \quad min$$

with constraints:

$$\sin(y-x-1) = 0$$
$$-2 \leq x \leq 2$$
$$-1.5 \leq y \leq 1.5$$

The theoretical solution of this problem is given by the point $x^* = (0.5, 1.5)$. We will solve this problem numerically, using Newton's method.

$$\mathbf{x_{n+1}} = \mathbf{x_n} - \mathbf{Hf}(\mathbf{x_n})^{-1} \cdot \nabla \mathbf{f}(\mathbf{x_n}) \tag{7}$$

Table: Some results of solving the optimization problem

Initial guess point (x , y)	Number of steps
(-1, 0)	6
(0 , 1)	2
(- 2 ,-1)	3
(π-2.5, -1.5)	6
(1.5 , 2.5 - π)	3
(1.8, 2.8 - π)	2

Conclusions.

- In Multicomplex-step differentiation approach there is no cancellation or subtraction error and one can make the truncation error negligibly small.
- Multicomplex-step differentiation was found to have very small approximation error while finite deference schemes suffer from round off errors
- Multicomplex-step differentiation takes shorter time to give solution of an optimization problem due to more accurate computations.
- Making one call of derivation procedure one is able to obtain different derivatives if required.

Thank You