EFG Financial Products

EFG Financial Products AG
Brandschenkestrasse 90
P.O. Box 1686, CH-8027 Zurich
Phone +41 58 800 1000, www.efgfp.com

Auto–Differentiation in Practice

Juergen Hakala

EFG Financial Products AG
Quant Analytics

14th Apr 2011

Disclaimer

- The views presented here are the authors' and do not necessarily reflect EFG FP's official policy or position, nor of any other organization the authors are or were affiliated with.

Agenda

- Autodifferentiation - What is it?
- Application of AD to model calibration
- Monte Carlo greeks using AD
- Conclusion

Auto–Differentiation

The concept

- Analytic differentiation of [suitable] functions is feasible
- Analytic integration of a general function is in general not possible
- Numerical integration of a general function is feasible
- Accurate numerical differentiation of general functions is difficult
- combine analytic differentiation with numerical differentiation to get more accurate results faster

Auto–Differentiation

The concept

Enhance the arithmetic operations from simple double calculus to dual numbers. The standard double variable is augmented by a dual part, similar to Clifford numbers (Clifford, 1873).

$$x \rightarrow (x\ ,\ \dot{x}) \tag{1}$$

The rules are

$$\begin{aligned}
(x\ ,\ \dot{x}) + (y\ ,\ \dot{y}) &= (x + y\ ,\ \dot{x} + \dot{y}) && (2)\\
(x\ ,\ \dot{x}) \cdot (y\ ,\ \dot{y}) &= (x \cdot y\ ,\ \dot{x} \cdot y + x \cdot \dot{y}) && (3)\\
(x\ ,\ \dot{x}) \,/\, (y\ ,\ \dot{y}) &= \left(x/y\ ,\ \frac{\dot{x}y - x\dot{y}}{y^2}\right) && (4)
\end{aligned}$$

Auto–Differentiation

The technical details

Define the standard mathematical functions for autodifferentiation variables, eg.

$$
\begin{aligned}
exp\,(x\ ,\ \dot{x}) &= (exp(x)\ ,\ exp(x)\dot{x}) && (5)\\
sin\,(x\ ,\ dotx) &= (sin(x)\ ,\ cos(x)\dot{x}) && (6)\\
& && (7)
\end{aligned}
$$

Complicated functions can be derived by composition of these basic expressions. The derivative is determined exactly.

Auto–Differentiation

Simple Example

$$
\begin{aligned}
f(x) &= xexp(-sin(x)^2/2) \\
f((x,\dot{x})) &= (x,\dot{x})exp(-sin(x,\dot{x})^2/2) \\
&= (x,\dot{x})exp(-(sin(x), cos(x))^2/2) \\
&= (x,\dot{x})exp(\,-\,(sin(x)^2, 2sin(x)cos(x))/2) \\
&= (x,\dot{x})exp((-sin(x)^2/2, -sin(x)cos(x))) \\
&= (x,\dot{x})\left(exp(-sin(x)^2/2), -sin(x)cos(x)exp(-sin(x)^2/2)\right) \\
&= \big(x\,\mathrm{exp}(-sin(x)^2/2), \\
&\quad -x\,\mathrm{sin}(x)cos(x)exp(-sin(x)^2/2) + exp(-sin(x)^2/2)\big)
\end{aligned}
\tag{8}
$$

Auto–Differentiation

Functions are compositions of atomic components

A computable function can be decomposed into a sequence of simple intermediate steps. The derivatives of these simple, atomic functions is known.

$$
\begin{aligned}
f(x) &= f_n \circ f_{n-1} \circ \cdots \circ f_1 \circ f_0(x) \\
f'(x) &= f'_n(f_{n-1} \circ \cdots \circ f_1 \circ f_0(x)) \\
&\quad \cdot f'_{n-1}(f_{n-2} \circ \cdots \circ f_1 \circ f_0(x)) \\
&\quad \cdots f'_1(f_0(x)) \cdot f'_0(x)
\end{aligned} \tag{9}
$$

Auto–Differentiation

The derivative through composition

is given as

$$f'(x) \quad = \quad f_n'(x_{n-1})f_{n-1}'(x_{n-2})\cdots f_1'(x_0) \tag{10}$$

setting x_n the variables after the execution of the first n functions, e.g. $x_1 = f_0(x)$. What is necessary is the computation and multiplication of the elemantary Jacobian matrices $f_n'(x_{n-1})$.

Auto–Differentiation

Forward mode

The computation from 0 to n

$$f'(x)dX = f'_n(x_{n-1})f'_{n-1}(x_{n-2})\cdots f'_1(x_0)dX \tag{11}$$

The results are passed via vector multiplications. Note that for each derivative a pass through the computation chain is needed.

Auto–Differentiation

Backward mode

The computation from n to 0

$$f'(x)dY = f_1'(x_0)\cdots f_{n-1}'(x_{n-2})f_n'(x_{n-1})dY \tag{12}$$

The results are passed down via vector multiplications. Note that intermediate Jacobians need to be stored when going through the computation chain. But all derivatives of a scalar function can be computed in one pass through the computation chain.

Auto–Differentiation

Higher order derivatives

- Higher order derivatives can be derived.
- Several passes through the computation chain are needed.
- Compute the Taylor expansion and derive the coefficients by substracting lower orders

Financial application – analytic formulas

- Use autodifferentiation on analytic formulas to derive super accurate greeks.

Financial application – analytic formulas

Example

- Barrier option delta, strike = 65, barrier = 90, t = 1
- Use a singularity/discontinuity avoidance for finite diff greeks

$$\partial f(x) \sim \frac{f(x+h) - f(x-h)}{2h}$$

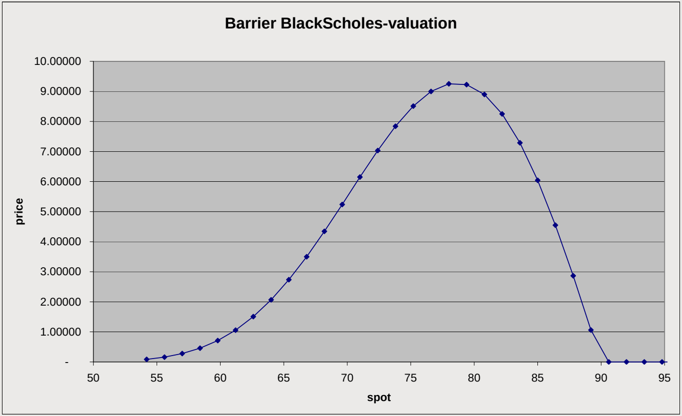
if f(x) is discontinuous in $[x, x+h]$ use

$$\partial f(x) \sim \frac{f(x) - f(x-2h)}{2h}$$

or even more accurately

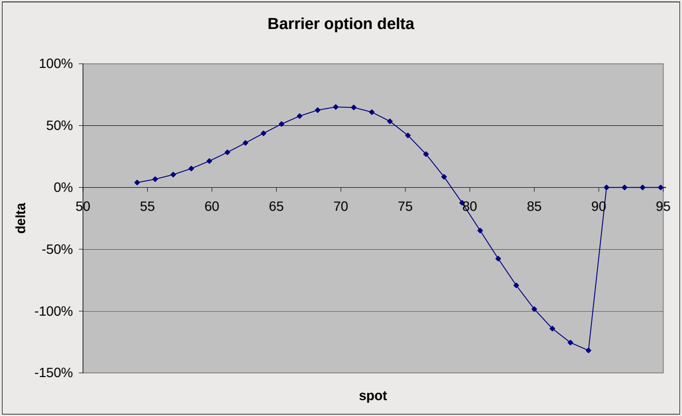
$$\partial f(x) \sim \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h}$$

Financial application – analytic formulas
Example



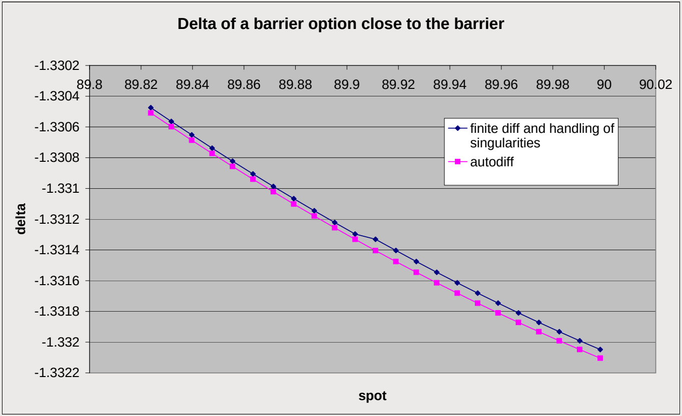
Financial application – analytic formulas

Example



Financial application – analytic formulas

Example



Auto–Differentiation

Calibration of models - differentiating the characteristic function

Optimize

$$\sum_i (P_i^{model}(\Theta) - P_i^{market})^2$$

- Least squares method (Levenberg–Marquardt)
- Use auto–diff for the determination of the Jacobian

Auto–Differentiation

Calibration of models - differentiating the characteristic function

Pricing with characteristic function, e.g. Heston model (Schoutens, 2006)

$$P = \frac{1}{2}(S_0 - e^{-rT}K) + \frac{1}{\pi}\int_0^\infty e^{rT} f_1(u) - K f_2(u) du \tag{13}$$

with

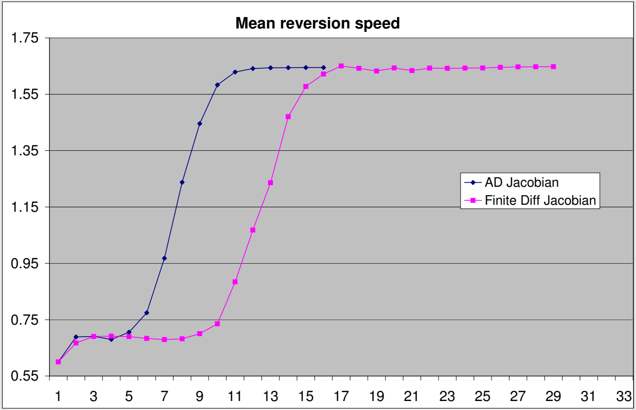
$$\begin{aligned} f_1 &= Re\left(\frac{e^{-iu\log(K)\phi(u-i,T)}}{iue^{rt}}\right) \\ f_2 &= Re\left(\frac{e^{-iu\log(K)\phi(u,T)}}{iu}\right) \end{aligned}$$

Auto–Differentiation

Financial application – characteristic function with integration

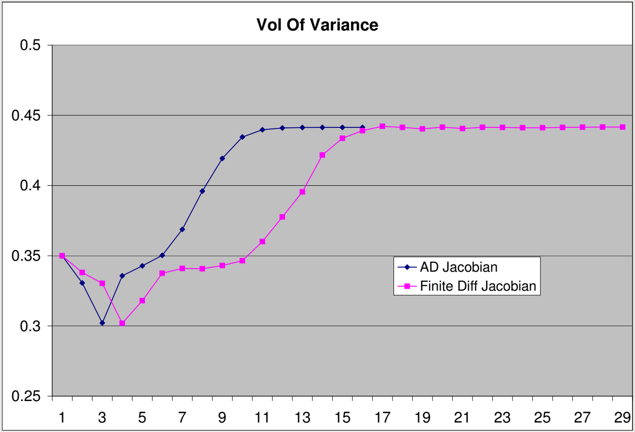
- Use autodifferentiation in vanilla pricing using characteristic functions
- e.g. Heston type formulas, jump–diffusion.
- Equip complex numbers with auto–differentiation
- Complex calculus does not need internal derivatives (with respect to a complex number)
- numerical integration algorithm with auto–differentiation

Financial application – Heston convergence
Example calibration



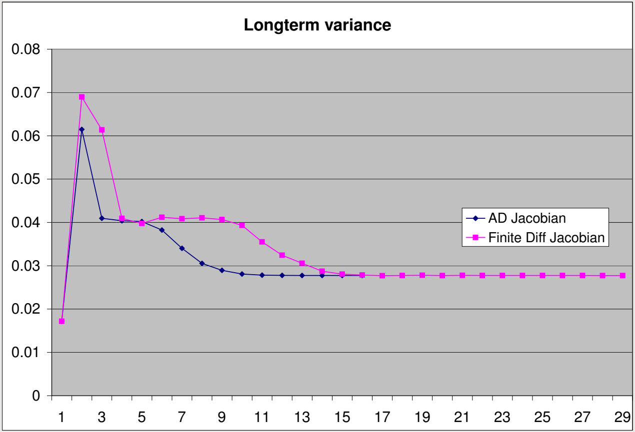
Financial application – Heston convergence

Example calibration



Financial application – Heston convergence

Example calibration



Auto–Differentiation

Financial application – characteristic function with integration

- Beneficial for calibration. Very accurate and fast derivatives with respect to the calibration params.
- Leads to improved convergence.
- Jacobian can be used in mapping back sensitivities to input parameter sensitivities.

Auto–Differentiation

Financial application – Monte Carlo

$$
\begin{aligned}
V &= \int P(X)p(X)dX \\
\frac{\partial V}{\partial \Theta} &= \int \partial_\Theta P(X)p(X)dX \\
\frac{\partial V}{\partial \Theta} &= \int P(X)\partial_\Theta p(X)dX = \int P(X)\partial_\Theta(\log(p(X)))p(X)dX
\end{aligned}
\tag{14}
$$

The expression $\partial_\Theta(\log(p(X)))$ is called weight, the method *Likelihood Ratio (LR)*. This method requires explicit knowledge of the transition density.

Auto–Differentiation
Financial application – Monte Carlo

How is the weight derived by auto-differentiation

$$\int P(X)p(X)dX = \int P(X)\frac{p(X_{AD})}{p(X)}p(X)dX \qquad (15)$$

Hence simulate $P(X)\frac{p(X_{AD})}{p(X)}$ under p to get the derivative on a per–path basis.

Auto–Differentiation
Financial application – Monte Carlo

Given state vector X_t and a given scheme, eg. Euler discretization

$$X_{t+1} = aX_t dt + \Sigma X_t dW \tag{16}$$

then the conditional probability can be expressed as

$$\begin{aligned} p(X_{t+1}|X_t) &= \frac{1}{(2\pi dt)^{n/2} det(\Sigma)^{1/2} \prod X_t^i} e^{-\frac{1}{2}\Psi^T \Sigma^{-1}(\Psi)} \\ \Psi &= \log(X_{t+1}) - \log(X_t) - adt \end{aligned} \tag{17}$$

.

Auto–Differentiation

Financial application – Monte Carlo

The derivative is computed using the implemented scheme and the joint probability is

$$p(X_{n+1}, X_n, \cdots, X_1 | X_0) = \prod_{t=1}^{n} p(X_{t+1} | X_t) \tag{18}$$

Auto–Differentiation

Financial application – Monte Carlo – Example

Lets consider the following payoff: A series of coupons is payed with $avg_i = N^{-1} \sum_j Perf_{i,j}$

$$\begin{aligned} C_i &= \max(C_{min}, \sum_j \min(\max(Perf_{i,j}, Perf_{min}), Perf_{max}) Ind(Perf_{i,j} > 0) \\ &+ avg_i Ind(Perf_{i,j} <= 0)) \end{aligned} \quad (19)$$

with performance defined as $Perf_{i,j} = (S_j(T_i)/S_j(T_0) - K_j)$.

Auto–Differentiation

Financial application – Monte Carlo – Example

Example with two assets, observations and coupon payments in 6M and 1Y, strike of 95% and a minimum performance of 1%, maximum performance of 6%, and a minimum coupon of 0.5%. Volatility for the assets is 12.5% respectively, and correlation 50%. Gamma (Asset 1) per 100.000:

	AD	2%	1%	0.5%	0.25%
1024	-32(327)	-1478(3238)	-2295(9851)	-1273(28889)	-15941(63163)
4096	-196(121)	-157(1376)	236(4504)	1313(16007)	9405(41731)
16384	-98(37)	-261(753)	-105(2749)	1409(5216)	6981(10832)
65536	-127(23)	-290(256)	-122(1011)	-1056(1998)	34(4125)
262144	-118(7)	-85(148)	18(335)	111(705)	1287(1952)
1048576	-114(2)	-96(48)	-85(125)	152(748)	1358(2457)

Auto–Differentiation

Financial application – Monte Carlo – Example

Volga (Asset 2) per 1 Million:

	AD	2%	1%	0.5%	0.25%
1024	310(139)	153(318)	157(1148)	227(1881)	672(3174)
4096	179(107)	245(139)	277(664)	937(1398)	1135(2135)
16384	249(37)	233(84)	303(203)	644(331)	767(868)
65536	210(16)	204(26)	186(97)	325(149)	320(723)
262144	210(9)	222(11)	230(34)	267(103)	278(371)
1048576	214(2)	222(7)	229(26)	264(78)	318(128)

Auto–Differentiation

Financial application – Monte Carlo – Local Vol Example

Local volatility on the same payoff:

Gamma (Asset 1) per 100.000:

	AD	1%
4096	-117(141)	-1748(4918)
16384	-89(67)	-714(1446)
65536	-125(24)	-779(854)
262144	-118(5)	-186(386)

Volga (Asset 2) per 1.000.000:

	AD	1%
4096	259(152)	-93(534)
16384	191(36)	159(212)
65536	211(17)	125(107)
262144	213(7)	201(42)

Auto–Differentiation

Financial application – Monte Carlo

- Impossible to use auto–differentiation on Monte Carlo directly.
- Instead use auto–differentiation on a path by path basis.
- Higher requirements on models, eg. local vol
- Drawback for LR method, eg. small volatilities and timesteps.
- Correlation sensitivity available

Auto–Differentiation

Financial application – Monte Carlo, discounting

Include the discounting for rate greeks

$$\begin{aligned} V &= \int \sum_i e^{-r_i T_i} P_i(X) p(X) dX \\ &= \sum_i e^{-r_i T_i} \int P_i(X) p(X) dX. \end{aligned} \tag{20}$$

Auto–Differentiation

Financial application – Monte Carlo, correlation

In FX, the correlation is usually derived using the cross–volatility

$$\rho_{XY,XZ} = \frac{\sigma_{XY}^2 + \sigma_{XZ}^2 - 2\sigma_{YZ}^2}{\sigma_{XY}\sigma_{XZ}} \tag{21}$$

From LR–scheme applied to the multi–asset FX simulation we get the sensitivity to two volatities and the correlation. To get the cross–vega and the correlation contribution of the non–cross vega we need to use the Jacobian of the expression above.

Auto–Differentiation
Conclusion

- Great method for lazy people
- Several areas of application
- Drawback: Legacy code and third party libraries
- libraries for AD are available and ready to be used: check out autodiff.org

Bibliography

- Michael B. Giles, *Monte Carlo evaluation of sensitivities in computational finance*, Oxford University Computing Laboratory, Report no. 7/12
- Hansjörg Albrecher, Philipp Mayer, Wim Schoutens, Jurgen Tistaert, *The Little Heston Trap*, (2006), Wilmott Magazine, January Issue, 83-92
- Christian P. Fries, Jörg Kampen, *Proxy Simulation Schemes for Generic Robust Monte-Carlo Sensitivities, Process Oriented Importance Sampling and High Accuracy Drift Approximation (With Applications to the LIBOR Market Model)*, (2005), Available at SSRN: http://ssrn.com/abstract=702642

Contact

EFG Financial Products AG
Jürgen Hakala
Quantitative Analytics
Brandschenkestrasse 90
P.O. Box 1686
8027 Zürich
Phone: +41 58 800 1028
email: juergen.hakala@efgfp.com
web: www.efgfp.com