Why do Quants love NAG?

a.k.a. NAG for Financial Engineering / Computational Mathematics

John Holden
john.holden@nag.co.uk
John Muddle
john.muddle@nag.co.uk

March 2016

UCL, Financial Computing and Analytics

nag®

Experts in numerical algorithms and HPC services

Agenda

- NAG Introduction
- NAG and UCL
- **Why do Quants love NAG?**
- **……..and why you should use NAG**
  - Problems in numerical computation
  - NAG's Numerical Libraries, Toolboxes and Tools

NAG Background

- Founded 1970
  - Not-for-profit organisation
  - Surpluses fund on-going R&D
- Mathematical and Statistical Expertise
  - Numerical Libraries of components
  - Consulting
- HPC Services
  - Computational Science and Engineering (CSE) support
  - Procurement advice, market watch, benchmarking

HPC Services

- Government, Academic and **Commercial**
- Full CSE service
  - Code porting, tuning, scaling, rewriting…
  - Training
  - 1-20 FTEs per annum
- Procurement advice/benchmarking

HECToR

ARM

CHPC
CENTRE FOR HIGH PERFORMANCE COMPUTING

HPC
WALES • CYMRU

nag

Portfolio

- Numerical Libraries
  - Highly flexible for use in many computing languages (C, C++, Fortran,…) programming environments, hardware platforms and for high performance computing methods
- Products for Excel®, MATLAB®, Python, .NET, and Java
  - Giving users of the spreadsheets and mathematical software packages access to NAG's library of highly optimized and often superior numerical routines
- Algorithmic Differentiation tools and services for C/C++, and Fortran (dco)
- NAG Fortran Compiler and GUI based Windows Compiler: Fortran Builder
- HPC Training and Consultancy services

University Site Licence

- Unlimited use of all NAG products
  - On Windows, Linux and Mac o/s
  - As long as for academic or research purposes
  - Installation may be on any university, staff or student machine
- Full access to NAG Support support@nag.co.uk
  - Request support or licence keys using your university e-mail address @ucl.ac.uk
  - Our software:
  - Includes online documentation - also www.nag.co.uk
  - Supplied with extensive example programs
    - data and results

NAG and UCL collaboration

- Performance improvements to Quantum Monte-Carlo code (CASINO) – working with Dario Alfe, Earth Sciences under HECToR
- GPU accelerated pivoted QR decomposition algo with BEM++ group - Tim Betcke et al, Maths
- Thomas Luu, 'Fast and accurate parallel computation of quantile functions for random number generation' (supervised by William Shaw, Maths)
- Long time partner for Financial Computing & Analytics CDT
  - and most recently with Eikon Appathon
- HPC and product training available
- New student projects and placements possible

THE ALAN TURING INSTITUTE

Bank of America Merrill Lynch

THOMSON REUTERS

Agenda

- NAG Introduction
- NAG and your University
- **Why do Quants love NAG?**
  - Problems in numerical computation
  - NAG's Numerical Libraries and Toolboxes
- NAG student prizes and jobs

Why bother?

- Numerical computation is difficult to do accurately
- Problems of
  - Overflow / underflow
    - How does the computation behave for large / small numbers?
  - Condition
    - How is it affected by small changes in the input?
  - Stability
    - How sensitive is the computation to rounding errors?
- Importance of
    - error analysis
    - information about error bounds on solution

Why Quants use NAG Libraries and Toolboxes?

- Global reputation for quality – accuracy, reliability and robustness…
- Extensively tested, supported and maintained code
- Reduces development time
- Allows concentration on your key areas
- Components
  - Fit into your environment
  - Simple interfaces to your favourite packages
- Regular performance improvements!
- Give "qualified error" messages e.g. tolerances of answers

Why you should use NAG?

- Site licence @ UCL
- Free for staff and students
- You can install on your personal machines
- Widely used in Industry
- Robust, reliable, accurate, fast and well tested
- Full access to NAG technical support team via support@nag.co.uk
  - But, we won't do your student assignments for you ;-)!

NAG Financial Maths Student Prizes

THE PRIZE: "Free attendance of finance conference including contribution to flights and accommodation"

THE REQUIREMENT:

- Good financial math "project" using NAG software
- Submit a description of the project, its goals and the use of NAG software

- Winning projects featured on NAG website

How to enter

Write to : nagmarketing@nag.co.uk

As sponsors of a wide variety of Mathematical Finance conferences and events we are currently able to offer the following choice of prizes:

- Global Derivatives, Trading & Risk Management
  - Budapest, Hungary 2016

Other prizes will be announced as conference tickets are secured.

An example: sample variance

For a collection of observations

$$\{x_i, i = 1,2,\dots,n\}$$

the mean is defined as

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and the variance as

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Example calculation

For this collection of observations

$$\{c-1, c, c+1\}$$

the mean is

$$\bar{x} = \frac{1}{3}(c-1+c+c+1) = c$$

and the variance is

$$s^2 = \frac{1}{2}[(-1)^2 + 0 + 1] = 1$$

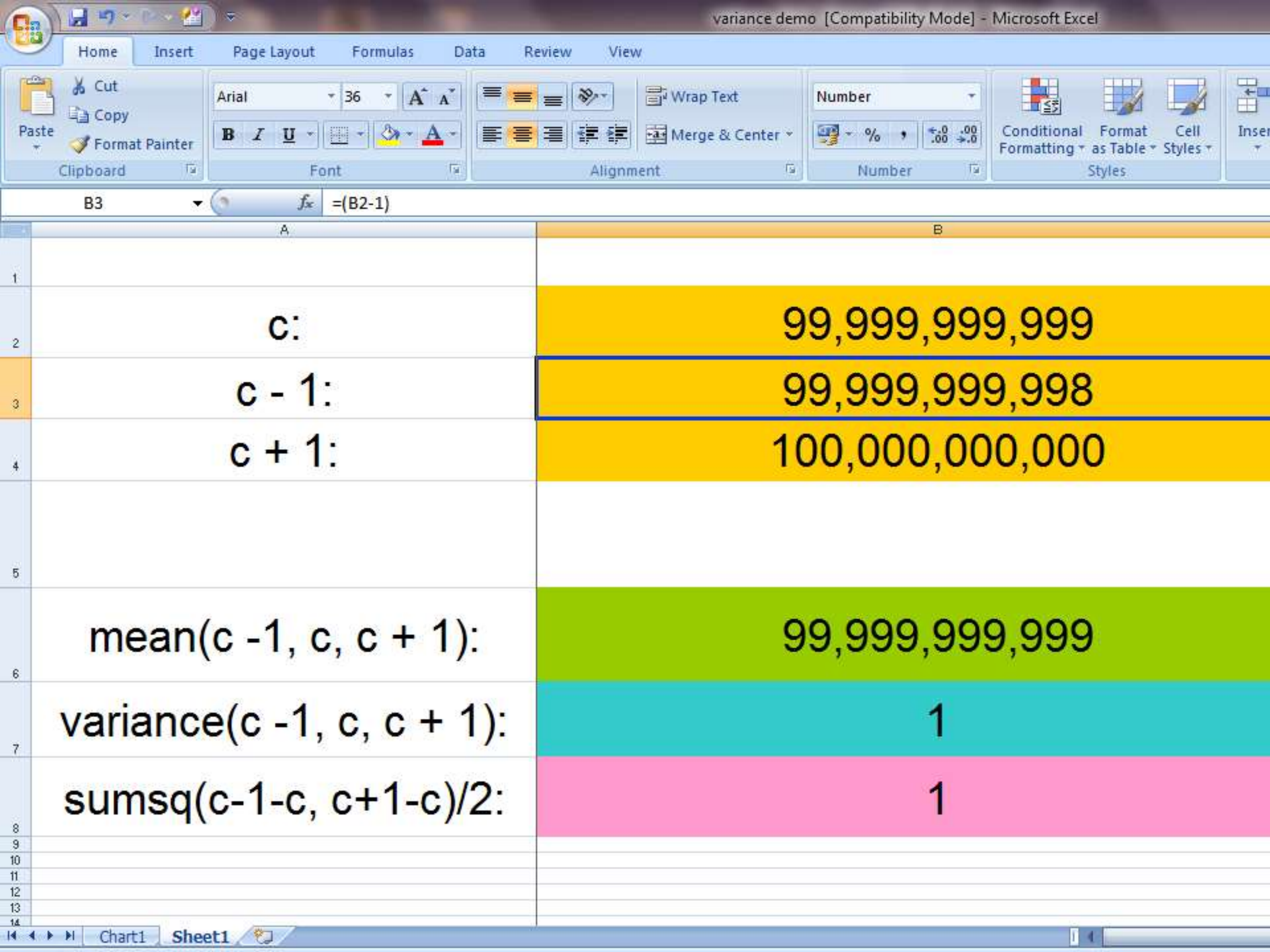
<Excel – variance demo>

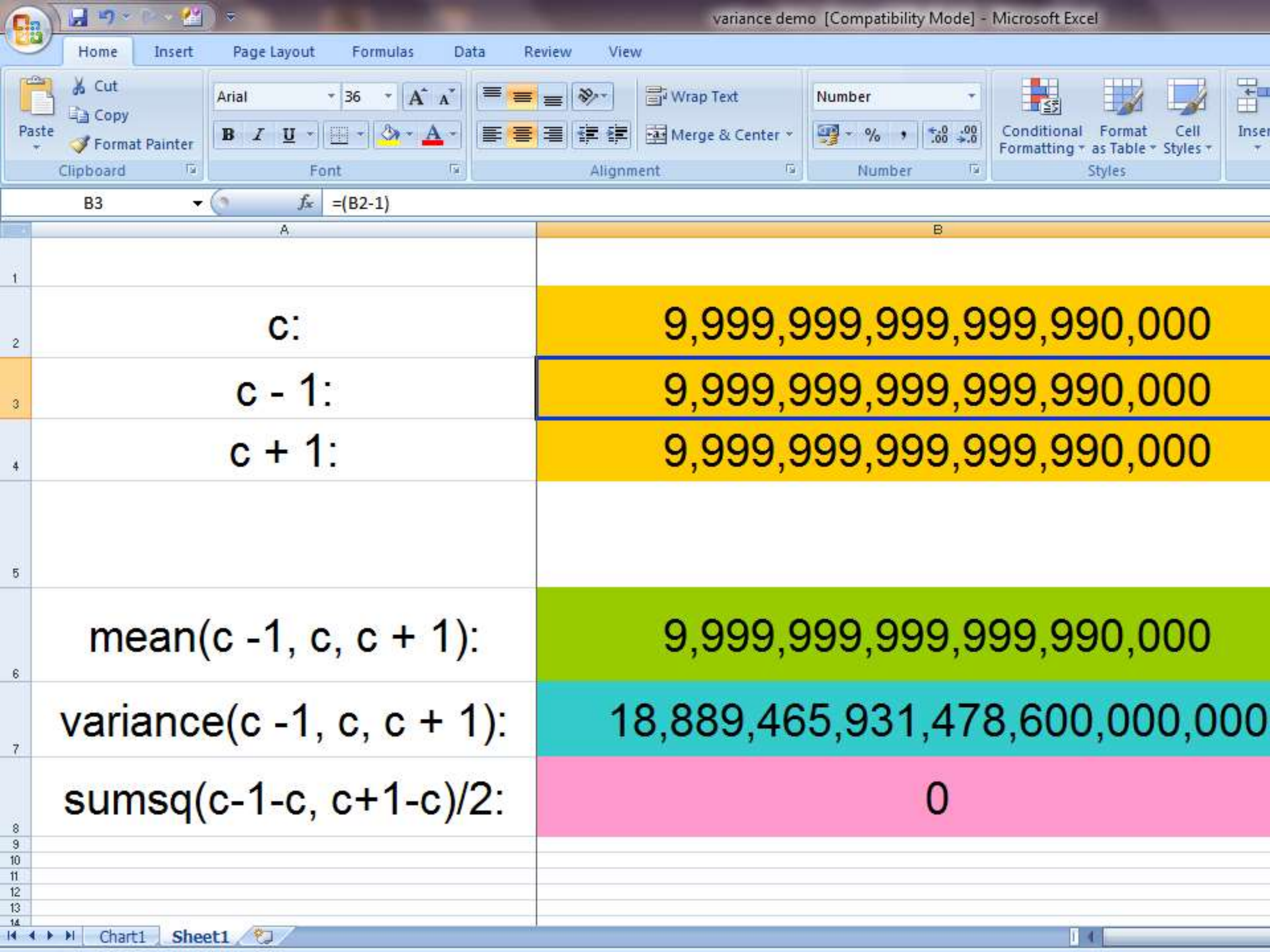
variance demo [Compatibility Mode] - Microsoft Excel

B3 f_x =(B2-1)

	A	B
1		
2	c:	0
3	c - 1:	-1
4	c + 1:	1
5		
6	mean(c -1, c, c + 1):	0
7	variance(c -1, c, c + 1):	1
8	sumsq(c-1-c, c+1-c)/2:	1

Chart1 Sheet1


variance demo [Compatibility Mode] - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
B3
=(B2-1)
c: 99,999,999,999
c - 1: 99,999,999,998
c + 1: 100,000,000,000
mean(c -1, c, c + 1): 99,999,999,999
variance(c -1, c, c + 1): 1
sumsq(c-1-c, c+1-c)/2: 1
Chart1
Sheet1


variance demo [Compatibility Mode] - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Paste
Cut
Copy
Format Painter
Clipboard
Arial
36
Font
Wrap Text
Merge & Center
Alignment
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
B3
=(B2-1)
A
B
c:
9,999,999,999,999,990,000
c - 1:
9,999,999,999,999,990,000
c + 1:
9,999,999,999,999,990,000
mean(c -1, c, c + 1):
9,999,999,999,999,990,000
variance(c -1, c, c + 1):
18,889,465,931,478,600,000,000
sumsq(c-1-c, c+1-c)/2:
0
Chart1
Sheet1

variance demo [Compatibility Mode] - Microsoft Excel

B3 =(B2-1)

c:	99,999,999,999,999,900,000
c - 1:	99,999,999,999,999,900,000
c + 1:	99,999,999,999,999,900,000
mean(c -1, c, c + 1):	99,999,999,999,999,900,000
variance(c -1, c, c + 1):	402,653,184
sumsq(c-1-c, c+1-c)/2:	268,435,456

Chart1 Sheet1

What's gone wrong?

Instead of

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Excel uses an (analytically identical) formula

$$s^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right)$$

- faster to calculate (one pass)
- accuracy problems if variance is small compared to X

from Finance - *k* Factor Problem

WWW.RISK.NET • NOVEMBER 2003 RISK 67

All your hedges in one basket

Leif Andersen, Jakob Sidenius and Susanta Basu present new techniques for single-tranche CDO sensitivity and hedge ratio calculations. Using factorisation of the copula correlation matrix, discretisation of the conditional loss distribution followed by a recursion-based probability calculation, and derivation of analytical formulas for deltas, they demonstrate a significant improvement in computational speeds

In a traditional synthetic collateralised debt obligation (CDO), the arranger tranches out credit losses on a pool of credit default swaps (CDSs) and passes them through to different investors. Assuming that investors for all tranches can be identified, the arranger is typically left with fairly moderate market exposure. For various reasons, placing the entire pool capital structure with investors has become increasingly difficult, and many recent credit basket derivatives expose the dealer to significant market risk. For instance, the recent 'single-tranche' CDO (STCDO) product involves the sale of a single CDO tranche to a single customer, leaving it to the arranger to manage the risk of the remaining capital structure. As STCDOs and similar 'custom' products offer significant customer benefits and are much less difficult to originate than traditional CDOs, such products are likely to increase in importance. This is especially true for managed trades where the customer has certain rights to alter the composition of the reference portfolio over time.

A basic prerequisite for active management of the risk of a credit basket derivative is the ability to accurately calculate the sensitivity of the security with respect to market and model parameters, most prominently the par CDS spreads of the underlying reference pool. The numbers of such sensitivities can be very large – many thousands – and can put considerable strain on computing resources. Moreover, the calculation of each of

where Q is the risk-neutral probability measure and λ_k is a (forward) default hazard rate function. The functions $p_k(T)$, $k = 1, \ldots, N$ can be bootstrapped by standard means from the quoted CDS spreads and are assumed known for all T.

Equation (1) fully establishes the risk-neutral marginal distribution of each default time τ_k. To construct the joint distribution of all default times, we here choose[4] to employ a Student-t copula, which we quickly define for reference. Defining vectors $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N)^T$ and $\mathbf{T} = (T_1, \ldots, T_N)^T$, the joint default time distribution in the Student-t copula, becomes:

$$Q(\boldsymbol{\tau} \le \mathbf{T}) = t_{N,\nu}\left(t_{1,\nu}^{-1}\left(p_1(T_1)\right), \ldots, t_{1,\nu}^{-1}\left(p_N(T_N)\right)\right) \quad (2)$$

where $t_{1,\nu}$ and $t_{N,\nu}$ are the one- and N-dimensional cumulative Student-t distribution functions with ν degrees of freedom, respectively. Recall that the density $\eta_{N,\nu}$ of an N-dimensional Student-t distribution with correlation matrix Σ is:

$$\eta_{N,\nu}(\mathbf{z}) = C_{N,\nu}\left(1 + \nu^{-1}\mathbf{z}^T\Sigma^{-1}\mathbf{z}\right)^{-\frac{\nu+N}{2}}, \quad C_{N,\nu} = \frac{\Gamma\left(\frac{\nu+N}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{|\Sigma|(\nu\pi)^N}} \quad (3)$$

where Γ is the gamma function. For high degrees of freedom, (3) approaches

from Finance - *k* Factor Problem

$$\min_{X \in \mathbb{R}^{n \times k}} f(X) := \|A - C(X)\|_F^2 \quad \text{subject to} \quad \sum_{j=1}^{k} x_{ij}^2 \leq 1.$$

All your hedges in one basket

Principal Factors method (Andersen et al., 2003) does NOT always converge to correct answer… (no convergence theory)

<u>Should have come to NAG….</u>

Our* spectral projected gradient method respects constraints, exploits convexity, converges to a feasible stationary point

**NAG Library G02AE - Borsdorf, Higham & Raydan, 2010,*

able strain on computing resources. Moreover, the calculation of each of

where Γ is the gamma function. For high degrees of freedom, (3) approaches

Numerical computation – DIY Vs NAG

- DIY implementations of numerical components have their place, but NOT in production code.
  - Handwritten and "hand me down" type code might be easy to implement, but it will…
    - NOT be well tested
    - NOT be fast
    - NOT be **stable**
    - NOT deliver good error handling
  - NAG implementations in contrast are **fast** and
    - Accurate
    - Well tested
    - Thoroughly documented
    - Give "qualified error" messages e.g. tolerances of answers (which the user can choose to ignore, but avoids proceeding blindly)

NAG provides the atomic bricks

- … for the quants to build the walls, houses and fancy castles!

- Users know NAG Components are here today, tomorrow and beyond
  - Functions are not removed when new ones added without sensible notice and advice
  - NAG functions are well documented

- Lets take a look….

NAG Library and Toolbox Contents

- Root Finding
- Summation of Series
- Quadrature
- Ordinary Differential Equations
- Partial Differential Equations
- Numerical Differentiation
- Integral Equations
- Mesh Generation
- Interpolation
- Curve and Surface Fitting
- Optimization
- Approximations of Special Functions
- Dense Linear Algebra
- Sparse Linear Algebra
- Correlation & Regression Analysis
- Multivariate Methods
- Analysis of Variance
- Random Number Generators
- Univariate Estimation
- Nonparametric Statistics
- Smoothing in Statistics
- Contingency Table Analysis
- Survival Analysis
- Time Series Analysis
- Operations Research

Use of NAG Software in Finance

- Portfolio analysis / Index tracking / Risk management
  - Optimization, linear algebra, copulas…
- Derivative pricing
  - PDEs, RNGs, multivariate normal, …
- Fixed Income/ Asset management / Portfolio Immunization
  - Operations research
- Data analysis
  - Time series, GARCH, principal component analysis, data smoothing, …
- Monte Carlo simulation
  - RNGs
- ……

Don’t take our word for it….

Financial Maths Professor and Morgan Stanley Consultant speeds up his optimization

Why an Exotics Options Quant loves NAG?

- General Problem
  - To build solvers for a variety of sophisticated financial models in a timely manner that are
    - robust,
    - stable,
    - quick
- Solution
  - Use robust, well tested, fast numerical components and tools
  - This allows the "expensive" quants to concentrate on the sophisticated models
    - avoiding distraction with low level numerical components

Problem 1: Finance PDE Solvers

- Major banks PDE solvers tend to be “proprietary” so there is a reluctance to take “end to end” solutions

- Several different numerical components needed
  - Sparse Linear Solvers
  - Functions for finding a range of Eigenvalues (stability checkers)
  - Special Linear Systems solvers (e.g. Banded Matrices)
  - ..

Problem 1: Finance PDE Solvers

- Major banks PDE solvers tend to be “proprietary” so there is a reluctance to take “end to end” solutions

  NAG to the rescue

- Several different numerical components needed
  - Sparse Linear Solvers ***√***
  - Functions for finding a range of Eigenvalues (stability checkers) ***√***
  - Special Linear Systems solvers (e.g. Banded Matrices) ***√***
  - .. ***√ √***

Problem 2: Calibration

- Major banks all need to calibrate their models

- Several different numerical components needed
  - Optimization functions (e.g. constrained non-linear optimizers)
  - Interpolation functions
  - Spline functions
  - ..

Problem 2: Calibration

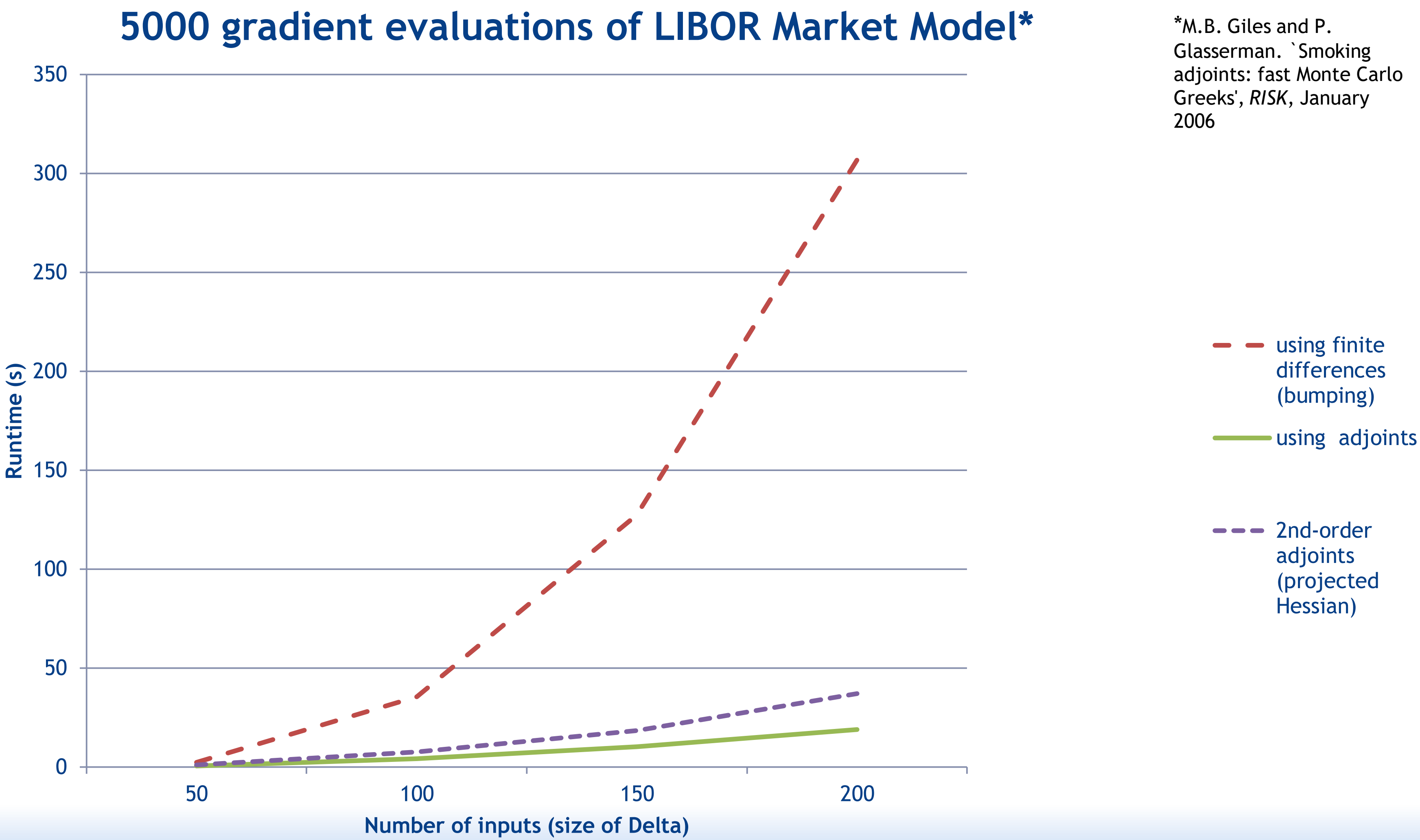
- Major banks all need to calibrate their models

  NAG to the rescue

- Several different numerical components needed
  - Optimisation functions (e.g. constrained non-linear optimisers) ***√***
  - Interpolation functions (used intelligently*) ***√***
  - Spline functions ***√***
  - .. ***√ √***

*interpolator must be used carefully –must know the properties to pick appropriate method

Problem 3: Why Adjoints?



*M.B. Giles and P. Glasserman. `Smoking adjoints: fast Monte Carlo Greeks', *RISK*, January 2006

3 Computing derivatives in finance is important…

- Calculating a product's sensitivities to a range of risk factors (a.k.a. Greeks) creates huge computational demand on risk and price models
- Traditional approach "bumping" - finite differences
  - Which is Computationally very expensive.. more hardware!

The alternatives to finite differences are

- Write derivative code by hand
  - Efficient, but difficult to write & highly error prone (need to develop original and adjoint models)
- Algorithmic Differentiation
  - flexible and only requires the original model - **obvious choice**

Problem 3: Why Adjoints

- Option pricing example from
- *Adjoint Algorithmic Differentiation Tool Support for Typical Numerical Patterns in Computational Finance* *Naumann & Du Toit*

  - finite difference method on 1,000 x 360 grid
  - wanted gradient of size n = 222
  - running on standard PC with 4GB of RAM

	Primal	Central FD	naïve AAD	robust AAD
run time (sec.)	0.5	175	∞	6

  - Delivering robust and efficient gradient code

NAG fits into your favourite environments

- Supporting Wide Range of Operating systems…
  - Windows, Linux, Mac
- …and a number of interfaces
  - C, C++,
  - Fortran,
  - VB, Excel & VBA,
  - C#, F#, VB.NET,
  - CUDA, OpenCL,
  - Hadoop & Apache SPARK
  - Java, Python
  - …
  - Excel,
  - LabVIEW,
  - MATLAB,
  - Maple,
  - Mathematica
  - R, S-Plus,
  - Scilab, Octave
  - …

NAG Code Examples

- Calling NAG from:
  - Excel® – NAG Fortran Library
  - Python – NAG4PY Wrapper
  - MATLAB® – NAG Toolbox for MATLAB®
- Examples @

github.com/numericalalgorithmsgroup

blog.nag.com

nag.co.uk/numeric/nagandexcel-examples

NAG Contacts / Keep in touch

http://www.nag.co.uk/edu/uk/ucl

Technical Support and Help

support@nag.co.uk

Your account manager

john.muddle@nag.co.uk

Join our group

NAGNews

http://www.nag.co.uk/NAGNews/Index.asp