Numerical software & tools for the actuarial community

John Holden
Jacques du Toit

11th September 2012

Actuarial Teachers’ and Researchers’ Conference
University of Leicester
Agenda

- NAG Introduction
- Software providers to the Insurance Market
- Numerical computation – why bother
  - Problems in numerical computation
  - NAG’s Numerical Libraries and Toolboxes
- Computational problems in Actuarial Science
Numerical Algorithms Group - What We Do

- NAG provides mathematical and statistical algorithm libraries widely used in industry and academia
- Established in 1970 with offices in Oxford, Manchester, Chicago, Taipei, Tokyo
- Not-for-profit organisation committed to research & development
- Library code written and contributed by some of the world’s most renowned mathematicians and computer scientists
- NAG’s numerical code is embedded within many vendor libraries such as AMD and Intel
- Many collaborative projects – e.g. CSE Support to the UK’s largest supercomputer, HECToR
Portfolio

- **Numerical Libraries**
  - Highly flexible for use in many computing languages, programming environments, hardware platforms and for high performance computing methods

- **Connector Products for Excel, MATLAB, .NET, R and Java**
  - Giving users of the spreadsheets and mathematical software packages access to NAG’s library of highly optimized and often superior numerical routines

- **NAG Fortran Compiler and GUI based Windows Compiler: Fortran Builder**

- **Visualization and graphics software**
  - Build data visualization applications with NAG’s IRIS Explorer

- **Consultancy services**
Software providers to the Insurance Market

- ACTUARIS
- AIR Worldwide
- Algorithmics
- Aon Benfield
- ARC
- AXIS
- Barrie & Hibbert
- BPS Resolver
- BWise
- ClusterSeven
- Conductor
- Conning
-...
-...

- Microsoft
- The Numerical Algorithms Group (NAG)
- Oracle Financial Services
- PolySytems
- RMS
- SAS Institute
- SunGard
- Towers Watson
- Trillium Software
- Ultimate Risk Solutions
- WySTAR
How is this software made?

- Do these software providers write all their own code?

- Do these software providers write all their own Numerical Code?

- Why not?
How is this software made?

- Do these software providers write all their own code?  
  No

- Do these software providers write all their own Numerical Code?  
  No

- Why not?  
  Let’s take a look
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
An example: sample variance

- For a collection of observations
  \[ \{x_i, i = 1 \ldots n\} \]
  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

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

<Excel – variance demo>
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td><strong>c:</strong></td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td><strong>c - 1:</strong></td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td><strong>c + 1:</strong></td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td><strong>mean(c -1, c, c + 1):</strong></td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td><strong>variance(c -1, c, c + 1):</strong></td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td><strong>sumsq(c-1-c, c+1-c)/2:</strong></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>(c:)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c - 1:)</td>
<td>99,999,999,999</td>
<td></td>
</tr>
<tr>
<td>(c + 1:)</td>
<td>100,000,000,000</td>
<td></td>
</tr>
<tr>
<td>(\text{mean}(c -1, c, c + 1):)</td>
<td>99,999,999,999</td>
<td></td>
</tr>
<tr>
<td>(\text{variance}(c -1, c, c + 1):)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(\text{sumsq}(c-1-c, c+1-c)/2:)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>9,999,999,999,999,990,000</td>
<td></td>
</tr>
<tr>
<td>c - 1</td>
<td>9,999,999,999,999,990,000</td>
<td></td>
</tr>
<tr>
<td>c + 1</td>
<td>9,999,999,999,999,990,000</td>
<td></td>
</tr>
<tr>
<td>mean(c -1, c, c + 1)</td>
<td>9,999,999,999,999,990,000</td>
<td></td>
</tr>
<tr>
<td>variance(c -1, c, c + 1)</td>
<td>18,889,465,931,478,600,000,000</td>
<td></td>
</tr>
<tr>
<td>sumsq(c-1-c, c+1-c)/2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>c:</td>
<td>99,999,999,999,999,999,900,000</td>
<td></td>
</tr>
<tr>
<td>c - 1:</td>
<td>99,999,999,999,999,999,900,000</td>
<td></td>
</tr>
<tr>
<td>c + 1:</td>
<td>99,999,999,999,999,999,900,000</td>
<td></td>
</tr>
<tr>
<td>mean(c -1, c, c + 1):</td>
<td>99,999,999,999,999,999,900,000</td>
<td></td>
</tr>
<tr>
<td>variance(c -1, c, c + 1):</td>
<td>402,653,184</td>
<td></td>
</tr>
<tr>
<td>sumsq(c-1-c, c+1-c)/2:</td>
<td>268,435,456</td>
<td></td>
</tr>
</tbody>
</table>
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 \( \bar{x} \)
Software providers to the Insurance Market

- ACTUARIS
- AIR Worldwide
- Algorithmics
- Aon Benfield
- ARC
- AXIS
- Barrie & Hibbert
- BPS Resolver
- B Wise
- ClusterSeven
- Conducter
- Conning
- ...
- ...

- ..
- ..
- Microsoft
- The Numerical Algorithms Group (NAG)
- Oracle Financial Services
- PolySytems
- RMS
- SAS Institute
- SunGard
- Towers Watson
- Trillium Software
- Ultimate Risk Solutions
- WySTAR
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 will...
    - NOT be well tested
    - NOT fast
    - NOT 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)
Why People 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!
NAG provides the atomic bricks

- ... for the domain specialists 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
NAG Data Mining Components

- **Data Cleaning**
  - Data Imputation
  - Outlier Detection

- **Data Transformations**
  - Scaling Data
  - Principal Component Analysis

- **Cluster Analysis**
  - k-means Clustering
  - Hierarchical Clustering

- **Classification**
  - Classification Trees
  - Generalised Linear Models
  - Nearest Neighbours

- **Regression**
  - Regression Trees
  - Linear Regression
  - Multi-layer Perceptron Neural Networks
  - Nearest Neighbours
  - Radial Basis Function Models

- **Association Rules**

- **Utility Functions**
  - To support the main functions and help with prototyping
NAG routines for GPUs

- Random Number Generators
  - L’Ecuyer mrg32k3a and Mersenne Twister (with skip-ahead) mt19937
    - Uniform distribution
    - Normal distribution
    - Exponential distribution
    - Support for multiple streams and sub-streams
  - Sobol sequence for Quasi-Monte Carlo (up to 50,000 dimensions)
  - Scrambled sequencing for Sobol (Hickernell)
  - Brownian Bridge
Traditional Uses of NAG Libraries

- NAG is used where non-trivial mathematics must be done quickly and accurately on computers

- Largest user groups (not in order)
  - Academic researchers (typically Statistics, Applied Mathematics, Finance, Economics, Physics, Engineering)
  - Engineers (fluid dynamics, large-scale PDE problems, simulations)
  - Statisticians (data mining, model fitting, analysis of residuals, time series, ...)
  - Quantitative analysts (asset modelling and risk analysis)
Use of NAG Software in Statistics

- **Multivariate Methods (G02/G04)**
  - Nearest correlation matrix, generalised regression with various error distributions (with and without missing data), robust/ridge/partial least squares regression, mixed effects and quantile regression, ...

- **Nonparametric Statistics (G08)**
  - Hypothesis testing

- **Survival Analysis (G12)**

- **Time Series Analysis (G13)**
  - SARIMA, VARMA, GARCH, with various modifications

- **Random Number Generators (G05)**
The NAG Library and Actuarial Statistics

- **Survival models:**
  - Cox regression model (g12bac)
  - Kaplan-Meier estimator (g12aac)
  - Weibull, exponential and extreme values (via g01gcc)

- **Risk analysis/loss functions:**
  - Distributions:
    - lognormal, gamma, beta etc. both distribution functions (g01) & random number generation (g05).

- **Other**
  - Time series (g05 and g13)
  - Convolutions: FFT's (c06)
  - Kernel density estimation
  - Graduation: generalised linear models (g02g)
  - Analysis of risk factors: generalised linear models (g02g)
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
Why Quantitative Analysts Love NAG?

- **General Problem**
  - To build asset models and risk engines in a timely manner that are
    - Robust
    - Stable
    - Quick

- **Solution**
  - Use robust, well tested, fast numerical components
  - This allows the “expensive” experts to concentrate on the modelling and interpretation
    - avoiding distraction with low level numerical components
Problem 1: Simulation (Monte Carlo)

- Simulation is important for scenario generation

- Several different numerical components needed
  - Random Number Generators
  - Brownian bridge constructor
  - Interpolation/Splines
  - Principal Component Analysis
  - Cholesky Decomposition
  - Distributions (uniform, Normal, exponential gamma, Poisson, Student’s t, Weibull,..)
  - ..
Problem 1: Simulation (Monte Carlo)

- Simulation is important for scenario generation

*NAG to the rescue (CPU or GPU)*

- Several different numerical components needed
  - Random Number Generators ✓
  - Brownian bridge constructor ✓
  - Interpolation/Splines ✓
  - Principal Component Analysis ✓
  - Cholesky Decomposition ✓
  - Distributions (uniform, Normal, exponential gamma, Poisson, Student’s t, Weibull,..) ✓
  - .. ✓ ✓
Problem 2: Calibration

- Financial institutions all need to calibrate their models

- Several different numerical components needed
  - Optimisation functions (e.g. constrained non-linear optimisers)
  - Interpolation functions
  - Spline functions
  - ..
Problem 2: Calibration

- Financial institutions 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: Historical VaR

- VaR methodology (important for identifying what variables might impact you most (eg Yen Vs USD))

- Several different numerical components needed
  - Time Series
  - Correlation and Regressions
  - Matrix functions
  - Cholesky Decomp
  - RNGs ..
Problem 3: Historical VaR

- VaR methodology (important for identifying what variables might impact you most (eg Yen Vs USD))

**NAG to the rescue**

- Several different numerical components needed
  - Time Series √
  - Correlation and Regressions √
  - Matrix functions √
  - Cholesky Decomp √
  - RNGs .. √ √
NAG fits into your favourite environments

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

Numerical Excellence in Finance
Numerical Excellence in Finance

NAG and Excel

Our libraries are easily accessible from Excel:

- Calling **NAG DLLs** using VBA
  - NAG provide VB Declaration Statements and Examples
  - NAG provide “Add-ins”
- Calling **NAG Library for .NET** using VSTO
- Functions with Reverse Communication (useful for Solver replication for example) can be provided
- Create **NAG XLLs**
NAG Portfolio Optimization in Excel

<table>
<thead>
<tr>
<th>covariance matrix</th>
<th>Aviva</th>
<th>Bp</th>
<th>Barclays</th>
<th>British Airways</th>
<th>Brit-Amer Tobacco</th>
<th>Lloyds</th>
<th>M&amp;S</th>
<th>Pearson</th>
<th>Tesco</th>
<th>Vodafone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aviva</td>
<td>0.0246</td>
<td>0.0054</td>
<td>0.0258</td>
<td>0.0275</td>
<td>0.0065</td>
<td>0.00291</td>
<td>0.0140</td>
<td>0.0061</td>
<td>0.0036</td>
<td>0.0043</td>
</tr>
<tr>
<td>Bp</td>
<td>0.0054</td>
<td>0.0092</td>
<td>0.0039</td>
<td>0.0020</td>
<td>0.0033</td>
<td>0.00377</td>
<td>0.0049</td>
<td>0.0017</td>
<td>0.0017</td>
<td>0.0034</td>
</tr>
<tr>
<td>Barclays</td>
<td>0.0258</td>
<td>0.0039</td>
<td>0.1031</td>
<td>0.0321</td>
<td>0.0215</td>
<td>0.0309</td>
<td>0.0083</td>
<td>0.0147</td>
<td>0.0032</td>
<td>0.0071</td>
</tr>
<tr>
<td>British Airways</td>
<td>0.0225</td>
<td>0.0020</td>
<td>0.0321</td>
<td>0.0736</td>
<td>0.0102</td>
<td>0.0328</td>
<td>0.0164</td>
<td>0.0174</td>
<td>0.0094</td>
<td>0.0068</td>
</tr>
<tr>
<td>Brit-Amer Tobacco</td>
<td>0.0065</td>
<td>0.0033</td>
<td>0.0021</td>
<td>0.0102</td>
<td>0.0068</td>
<td>0.0093</td>
<td>0.0164</td>
<td>0.0208</td>
<td>0.0037</td>
<td>0.0071</td>
</tr>
<tr>
<td>Lloyds</td>
<td>0.0291</td>
<td>0.0059</td>
<td>0.0377</td>
<td>0.0328</td>
<td>0.0093</td>
<td>0.0464</td>
<td>0.0144</td>
<td>0.0051</td>
<td>0.0034</td>
<td>0.0071</td>
</tr>
<tr>
<td>M&amp;S</td>
<td>0.0140</td>
<td>0.0049</td>
<td>0.0309</td>
<td>0.0184</td>
<td>0.0225</td>
<td>0.0421</td>
<td>0.0103</td>
<td>0.0087</td>
<td>0.0044</td>
<td>0.0068</td>
</tr>
<tr>
<td>Pearson</td>
<td>0.0061</td>
<td>0.0017</td>
<td>0.0083</td>
<td>0.0138</td>
<td>0.0051</td>
<td>0.0103</td>
<td>0.0084</td>
<td>0.0046</td>
<td>0.0011</td>
<td>0.0071</td>
</tr>
<tr>
<td>Tesco</td>
<td>0.0096</td>
<td>0.0017</td>
<td>0.0147</td>
<td>0.0174</td>
<td>0.0138</td>
<td>0.0087</td>
<td>0.0113</td>
<td>0.0113</td>
<td>0.0057</td>
<td>0.0071</td>
</tr>
<tr>
<td>Vodafone</td>
<td>0.0043</td>
<td>0.0034</td>
<td>0.0032</td>
<td>0.0094</td>
<td>0.0037</td>
<td>0.0068</td>
<td>0.0034</td>
<td>0.0051</td>
<td>0.0164</td>
<td>0.0071</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Average Return</th>
<th>Aviva</th>
<th>Bp</th>
<th>Barclays</th>
<th>British Airways</th>
<th>Brit-Amer Tobacco</th>
<th>Lloyds</th>
<th>M&amp;S</th>
<th>Pearson</th>
<th>Tesco</th>
<th>Vodafone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Deviation</td>
<td>-0.0082</td>
<td>0.0152</td>
<td>-0.0070</td>
<td>0.0483</td>
<td>0.0706</td>
<td>-0.0636</td>
<td>0.0191</td>
<td>0.0196</td>
<td>0.0452</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>0.1568</td>
<td>0.0907</td>
<td>0.3211</td>
<td>0.2714</td>
<td>0.0627</td>
<td>0.2154</td>
<td>0.2051</td>
<td>0.0515</td>
<td>0.1064</td>
<td>0.1260</td>
</tr>
</tbody>
</table>

Desired portfolio return: 6.00%

Perform portfolio optimization for the desired return.

No short selling
Risk: 7.07%
Structure of portfolio:

<table>
<thead>
<tr>
<th>Short sell allowed</th>
<th>Aviva</th>
<th>Bp</th>
<th>Barclays</th>
<th>British Airways</th>
<th>Brit-Amer Tobacco</th>
<th>Lloyds</th>
<th>M&amp;S</th>
<th>Pearson</th>
<th>Tesco</th>
<th>Vodafone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk</td>
<td>6.35%</td>
<td>6.22%</td>
<td>9.79%</td>
<td>4.10%</td>
<td>12.20%</td>
<td>52.77%</td>
<td>-15.52%</td>
<td>16.28%</td>
<td>26.05%</td>
<td>0.81%</td>
</tr>
</tbody>
</table>
Example – Kaplan-Meier survival probabilities

- **Data**
  - Life tables for WHO Member States
    - Global level of child and adult mortality
  - How many people out of 100,000 die at birth, until 1YO, 5YO, etc. and how many people live 100 years or longer
Input ... and ... output ...

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Censored</td>
<td>Frequency</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>4,380</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1,649</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>544</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>379</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>630</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>953</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>1,124</td>
</tr>
<tr>
<td>9</td>
<td>30</td>
<td>1,355</td>
</tr>
<tr>
<td>10</td>
<td>35</td>
<td>1,605</td>
</tr>
<tr>
<td>11</td>
<td>40</td>
<td>2,032</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>2,784</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>3,928</td>
</tr>
<tr>
<td>14</td>
<td>55</td>
<td>5,323</td>
</tr>
<tr>
<td>15</td>
<td>60</td>
<td>7,248</td>
</tr>
<tr>
<td>16</td>
<td>65</td>
<td>9,577</td>
</tr>
<tr>
<td>17</td>
<td>70</td>
<td>12,056</td>
</tr>
<tr>
<td>18</td>
<td>75</td>
<td>13,511</td>
</tr>
<tr>
<td>19</td>
<td>80</td>
<td>13,240</td>
</tr>
<tr>
<td>20</td>
<td>85</td>
<td>10,195</td>
</tr>
<tr>
<td>21</td>
<td>90</td>
<td>5,211</td>
</tr>
<tr>
<td>22</td>
<td>95</td>
<td>1,780</td>
</tr>
<tr>
<td>23</td>
<td>100</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Survival Probability</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>5</td>
</tr>
<tr>
<td>43</td>
<td>5</td>
</tr>
<tr>
<td>44</td>
<td>10</td>
</tr>
<tr>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>46</td>
<td>15</td>
</tr>
<tr>
<td>47</td>
<td>15</td>
</tr>
<tr>
<td>48</td>
<td>20</td>
</tr>
<tr>
<td>49</td>
<td>20</td>
</tr>
<tr>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>51</td>
<td>25</td>
</tr>
<tr>
<td>52</td>
<td>30</td>
</tr>
<tr>
<td>53</td>
<td>30</td>
</tr>
<tr>
<td>54</td>
<td>35</td>
</tr>
<tr>
<td>55</td>
<td>35</td>
</tr>
<tr>
<td>56</td>
<td>40</td>
</tr>
<tr>
<td>57</td>
<td>40</td>
</tr>
<tr>
<td>58</td>
<td>45</td>
</tr>
<tr>
<td>59</td>
<td>45</td>
</tr>
<tr>
<td>60</td>
<td>50</td>
</tr>
<tr>
<td>61</td>
<td>50</td>
</tr>
<tr>
<td>62</td>
<td>55</td>
</tr>
<tr>
<td>63</td>
<td>55</td>
</tr>
<tr>
<td>64</td>
<td>60</td>
</tr>
<tr>
<td>65</td>
<td>60</td>
</tr>
<tr>
<td>66</td>
<td>65</td>
</tr>
<tr>
<td>67</td>
<td>70</td>
</tr>
<tr>
<td>68</td>
<td>70</td>
</tr>
<tr>
<td>69</td>
<td>75</td>
</tr>
<tr>
<td>70</td>
<td>75</td>
</tr>
<tr>
<td>71</td>
<td>80</td>
</tr>
<tr>
<td>72</td>
<td>80</td>
</tr>
<tr>
<td>73</td>
<td>85</td>
</tr>
<tr>
<td>74</td>
<td>85</td>
</tr>
<tr>
<td>75</td>
<td>90</td>
</tr>
<tr>
<td>76</td>
<td>90</td>
</tr>
<tr>
<td>77</td>
<td>90</td>
</tr>
<tr>
<td>78</td>
<td>90</td>
</tr>
<tr>
<td>79</td>
<td>95</td>
</tr>
<tr>
<td>80</td>
<td>95</td>
</tr>
<tr>
<td>81</td>
<td>100</td>
</tr>
</tbody>
</table>
... followed by Excel plot
How do you call the functions in Excel
How do you call the functions in Excel

Enter the inputs from the spreadsheet

Excel function wizard
What’s under the hood?

NAG Library function called via VBA
NAG and .NET

NAG solutions for .NET

1. Call NAG C (or Fortran) DLL from C#

2. NAG Library for .NET
   
   “a more natural solution”
   
   - DLL with C# wrappers
   - Integrated help
   - Not yet the full Library, but most widely used chapters included.

Very popular with .NET dev community inc. in Financial Services.
NAG Toolbox for MATLAB

- Contains essentially all NAG functionality
  - not a subset
- Runs under Windows (32/64bit), Linux (32/64-bit) and Mac (64 bit).
- Comprehensive documentation (in MATLAB and pdf)
- Easy migration to production code in C/C++ or Fortran
- Can be used with MATLAB compiler
1 Purpose

d03eb uses the Strongly Implicit Procedure to calculate the solution to a system of simultaneous algebraic equations of five-point molecule form on a two-dimensional topologically-rectangular mesh. (Topological means that a polar grid, for example (r, θ), can be used, being equivalent to a rectangular box.)

2 Syntax

\[ [t, itconm, itused, resids, chmps, ifail] = d03eb(nd1, a, b, d, e, q, t, aparam, imax, iflag, mnk, n, nd2) \]

3 Description

Given a set of simultaneous equations

\[ \begin{align*}
    Mq &= \mathbf{b} \\
    \mathbf{q} &= \mathbf{q_{in}}
\end{align*} \]

(which could be nonlinear), for example, from a finite difference representation of a two-dimensional elliptic partial differential equation and its boundary conditions, the routine determines the values of the dependent variables \( t \) and \( q \) in a known vector of length \( n_1 \times n_2 \) and is a (\( n_1 \times n_2 \)) by (\( n_1 \times n_2 \)) matrix.

The equations must be in five-diagonal form.

\[ \begin{align*}
    a_{ij}q_{j} &= b_i, \\
    \text{for } i = 1, \ldots, n_1, j = 1, \ldots, n_2, \text{ provided } c_{ij} \neq 0. \text{ Indeed, if } c_{ij} = 0, \text{ then the equation is assumed to be } q_i = q_{in}.
\end{align*} \]

For example, if \( n_1 = 3 \) and \( n_2 = 2 \), the equations take the form:

\[ \begin{align*}
    a_1q_1 + a_2q_2 + a_3q_3 &= b_1, \\
    a_4q_1 + a_5q_2 + a_6q_3 &= b_2, \\
    a_7q_1 + a_8q_2 + a_9q_3 &= b_3
\end{align*} \]
NAG Toolbox for MATLAB

- Offers complementary functionality to MATLAB
- Alternative to several specialist toolboxes
- “I really like the NAG Toolbox for MATLAB for the following reasons (among others):
  - It can speed up MATLAB calculations – see my article on MATLAB’s interp1 function for example.
  - Their support team is superb.”

- Senior Developer
  - “concerning the ‘nearest correlation’ algorithm. I have to say, it is very fast, it uses all the power of my pc and the result is very satisfactory.”
Computational problems in Actuarial Science

- Liability Modelling
- Asset Modelling
- Solvency II
- Nearest Correlation Matrix example
"Traditional" actuarial science is focused predominantly on liability modelling

- Forecast cash flows directly linked to mortality/longevity
- Example: a pension scheme. Premiums received until retirement, pension paid until death, lump sum paid upon death
- Requires *some* modelling of market conditions (assumptions on inflation, gilt yields, index returns, ...)
- Fair to say often this modelling is not very sophisticated and is not very computationally demanding.
Asset Modelling

- Liability modelling very well understood
  - Been doing it for more than two centuries, mostly get it right (sometimes get it wrong)
  - Commercial packages to do this (Prophet, MoSes, ...)
  - Often heavily regulated (e.g. pensions)

- Asset modelling, in the actuarial context, perhaps less so
  - Traditional view been to model average behaviour over long horizons – simplicity is sensible, since so many assumptions anyway
Solvency II

- There is a regulatory push to change this
  - “Solvency II = Basel for insurers”
  - Similar risk methodology as banks, being introduced for insurers
  - Aim is to stress insurer’s balance sheet to various shocks, especially market shocks

- Requires more explicit modelling of assets
  - Initial guidelines laid down by regulator – pretty simplistic
  - However, as with Basel, insurers encouraged to develop own approaches (which would be less punitive)
  - Horizons fairly short-dated
Asset modelling is difficult!
- Ask a financial mathematician (or a quant)
- Technically demanding and computationally demanding

Moreover, every market is unique
- Not just across asset classes, but different countries as well. No “one-size-fits-all” approach possible
- Each has own behaviour, own peculiarities

NAG Library used extensively for building sophisticated, robust asset models and risk engines
NEAREST CORRELATION MATRIX
Mathematically, a correlation matrix $C \in \mathbb{R}^{n \times n}$ is ...

1. Square
2. Symmetric with ones on diagonal
3. Is positive semi-definite: $x^T C x \geq 0$ for all $x \in \mathbb{R}^n$

How do we estimate correlations?

- Historical data
- Parametric methods such as Gaussian Copulas
- Try to back it out from options markets

Typically 1 and 2 easy enough to ensure

- Ensuring positive semi-definite can be tricky
Example: Nearest Correlation Matrix

- **Historical data**
  - Take time series for several observables and try to estimate correlation

- **Gaussian copula**
  - Model for turning a set of marginals + a correlation matrix into a joint distribution.
  - Was popular in credit modelling until 2008/9 proved it was (in many cases) wholly inadequate

- **Infer from options markets**
  - Combine individual options and options on indexes to back out correlation structure
Example: Nearest Correlation Matrix

- In all these cases, need to work with correlation matrices estimated from “real world” data
  - Real data is messy

- Given importance of correlation, what happens if estimate not mathematically correct?
NAG Library can find the “nearest” correlation matrix to a given square matrix $A$

- G02AA solves the problem $\min_C \| A - C \|_F^2$ in Frobenius norm
- G02AB incorporates weights $\min_C \| W^{1/2} (A - C) W^{1/2} \|_F^2$
- Weights useful when have more confidence in accuracy of observations for certain observables than for others
Example: Nearest Correlation Matrix

The effect of $W$:

$$\mathbf{A} = \begin{bmatrix} 0.4218 & 0.6557 & 0.6787 & 0.6555 \\ 0.9157 & 0.3571 & 0.7577 & 0.1712 \\ 0.7922 & 0.8491 & 0.7431 & 0.7060 \\ 0.9595 & 0.9340 & 0.3922 & 0.0318 \end{bmatrix}$$

$W = \text{diag}(\{10,10,1,1\})$

$$W \mathbf{A} W^T = \begin{bmatrix} 42.1761 & 65.5741 \\ 91.5736 & 35.7123 \end{bmatrix} \begin{bmatrix} 6.7874 & 6.5548 \\ 7.5774 & 1.7119 \end{bmatrix} \begin{bmatrix} 7.9221 & 8.4913 \\ 9.5949 & 9.3399 \end{bmatrix}$$

Whole rows/cols weighted by $w_i$

Elements weighted by $w_i \times w_j$
Can also do dimension reduction (G02AE)

- So-called factor models. Similar to PCA in regression
- Suppose have assets $Y_1, \ldots, Y_n$, an $n$ dimensional source of noise $W$ and an $n \times n$ “correlation” matrix $A$ where

$$
\begin{bmatrix}
Y_1^1 \\
\vdots \\
Y_n^n
\end{bmatrix} = F_t + A
\begin{bmatrix}
W_1^1 \\
\vdots \\
W_n^n
\end{bmatrix}
$$

- For example, a simple multi-asset model: one factor (e.g. Brownian motion) for each asset, and $AA^T$ gives correlation between all factors
3\textsuperscript{rd} Example: Nearest Correlation Matrix

- Can use NAG Library to \textit{reduce the number of factors}
  - Find a $n \times k$ matrix $D$ (where $k < n$) such that
    \[
    \begin{bmatrix}
    Y_1^t \\
    \vdots \\
    Y_n^t
    \end{bmatrix}
    = F_t + D
    \begin{bmatrix}
    W_1^t \\
    \vdots \\
    W_k^t
    \end{bmatrix}
    \]
  - Crucially, $DD^T$ gives a correlation structure as close as possible to the original structure implied by $A$
  - Can be very useful to reduce complexity and computational cost of some models and applications
Example: Nearest Correlation Matrix

- Spreadsheet (if time)
NAG is a HPCFinance partner

High Performance Computing in Finance

http://www.hpcfinance.eu

The network is recruiting for

- Early Stage Researchers (ESRs ~ PhD Students)
- Experienced Researchers (ERs ~ Post Docs)
NAG and Actuarial Science - Summary

- NAG is keen to collaborate in building actuarial models and risk engines
  - Your requirements likely to be different from banks/hedge funds
  - We want to make sure we have what you need
- Risk engines likely to involve a LOT of computation
  - NAG has *significant* experience in HPC services, consulting and training
  - We know how to do large scale computations efficiently
  - *This is non-trivial!* Our expertise has been sought out and exploited by organisations such as (BP, HECToR, Microsoft, Oracle, Rolls Royce, .......)
Keep in touch

Many of you are already licensed to use NAG software....

Technical Support and Help

support@nag.co.uk

To reach the speakers

john.holden@nag.co.uk
jacques@nag.co.uk

NAGNews

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