Using NAG Numerical Software via C, C++, Excel, Fortran, MATLAB & other environments

LTCC

John Holden, David Sayers, Louise Mitchell

Results Matter. Trust NAG.
Agenda...

- Introduction to NAG
- Technical overview
  - A few examples..
    - NAG Fortran / C Library for Windows
    - NAG and Excel
    - NAG Toolbox for MATLAB
    - Fortran Builder (NAG’s New Windows Fortran Compiler)
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 and 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
Partnerships with leading academics

- University of Aachen
  - Uwe Naumann
- K.U. Leuven
  - Wim Schoutens
- University of Manchester
  - Peter Duck, Nick Higham, Ser Huang Poon, ..
- University of Oxford
  - Mike Giles,
- University of Vienna
- Stanford University
Portfolio

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

- **Connector Products for MATLAB and Maple**
  - Giving users of the mathematical software packages MATLAB and Maple access to NAG’s library of highly optimized and often superior numerical routines

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

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

- **Consultancy services**
Why Use NAG Maths Libraries?

- Global reputation for quality – accuracy, reliability and robustness…
- Extensively tested, supported and maintained code
- Reduce development time
- Concentrate on your key areas
- Components
  - Fit into your environment
  - Simple interfaces to your favourite packages
- Regular performance improvements!
What does the NAG / LTCC licence cover?

- See LTCC website for “up to date” information / product listing. [or contact NAG via operations@nag.co.uk quoting ref: NAG/LTCC/JCH]

- Unlimited use for the licensed implementations
  - As long as for academic or research purposes
  - Installation may be on any university, staff or student machine as long as they are from the dept or site

- Full access to NAG Support support@nag.co.uk

- Our software:
  - Includes online documentation - also www.nag.co.uk
  - Supplied with extensive example programs
    - data and results
Technical Agenda

- The NAG Engine
- Algorithmic contents
- Ease of Integration
  - NAG and Excel examples
  - Navigating around the NAG toolbox in MATLAB
- NAG Optimisation Chapters
- Next release
  - Option Pricing Functions
The NAG Engine

NAG software is based on NAG Engine technology

User-callable library routines are thin wrappers

NAG C Library

NAG Fortran Library

NAG Toolbox for MATLAB

NAG Engine (algorithmic repository)

NAG SMP Library

Other NAG Software
NAG Library 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 and Regression Analysis
- Multivariate 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 Libraries – a quick introduction

- NAG Fortran Library
  - C:\Program Files\NAG\FL21
    - Manual – html – Mk21
    - Samples – surface fit
    - Optimisation E04 chapter introduction

- NAG C Library

  C:\Program Files\NAG\CL08\cldll084zl\projects

N.B. Manual needs to be installed separately
NAG Libraries Ease of Integration

- C++ (various)
- C# / .NET
- Visual Basic
- Java
- Borland Delphi
- F#
- Python
- ...
- ...
- and more

- Excel
- MATLAB
- Maple
- LabVIEW
- R and S-Plus
- SAS
- Simfit
- ...
- and more
NAG and Excel..

www.nag.co.uk/numeric/callingDLLsfromotherlang.asp

<start Excel here>
NAG and Excel..

- **Our libraries are easily accessible from Excel**
  - Calling DLLs using VBA
  - NAG provide VB Declaration Statements and Examples

- **Excel Add-ins**
  - NAG’s Statistical Add-in for Excel
    - Sophisticated Add-in offering 76 statistical functions
    - Function/array “driven”
  - NAG Schools Excel Add-in (N-SEA)
    - Basic statistical functions including graphs
    - Menu Drive

=start Excel here>
Maple-NAG Connector

- Works with “latest” versions of:
  - Maple 10, 11 & 12
  - NAG C Library 7 & 8
  - The connector supports Mark 7 functionality

- Runs under:
  - Mac (PowerPC, Intel Mac - 32-bit)
  - Linux (32-bit)
  - Windows (32-bit)

<start Maple here>
NAG Toolbox for MATLAB

http://www.nag.co.uk/numeric/MB/start.asp

<start MATLAB here>   <doc, G01aa.., D01AJ>
NAG Toolbox for MATLAB

- Built as MATLAB mex files
  - Auto-generated from XML documentation
- Contains essentially all NAG functionality
  - not a subset
- Currently runs under Windows (32/64bit) or Linux (32/64-bit).
- Installed under the usual MATLAB toolbox directory
- Makes use of a DLL or shared version of the NAG Library
- Can be used with MATLAB compiler

<start MATLAB here> <doc, G01aa.., D01AJ>
Chapter e04 – Minimization / Maximization

Problem:  \( \text{minimize } F(x_1, x_2, \ldots, x_n) \)

possibly subject to constraints

The function \( F(x) \) is called the \textit{objective function}.  We wish to determine \( x \), the \( n \)-vector of variables.

May have:

- No constraints
- Bound constraints: \( l_i \leq x_i \leq u_i \)
- Linear or nonlinear constraints: \( l \leq G(x) \leq u \)
Unconstrained optimization
Linearly constrained optimization

Feasible Region
Nonlinear constraints

Feasible Region
Chapter e04

Problems categorized according to properties of objective function:

- nonlinear
- sum of squares of nonlinear functions
- quadratic
- linear

Example – nonlinear objective and constraints:

Minimize \( f(x,y) = (1 - x)^2 + 100(y - x^2)^2 \)

subject to \( x^2 + y^2 \leq 2 \)
\( -2 \leq x \leq 2 \)
E04WD

- Sequential quadratic programming (SQP) algorithm
  - obtains search directions from a sequence of QP subproblems.
  - designed for problems with many variables and constraints
  - P. Gill (San Diego), W. Murray (Stanford) and M. Saunders (Stanford)
Chapter e04

It is important to choose a method appropriate to your problem type, for efficiency and the best chance of success.

NAG documentation is comprehensive – for advice see the Chapter Introduction for e04:

www.nag.co.uk/numeric/FL/manual/pdf/E04/e04_intro.pdf
www.nag.co.uk/numeric/CL/nagdoc_cl08/pdf/E04/e04_intro.pdf

<run rosenbrock_sd_demo, rosenbrock_sqp_demo, rosenbrock_lsq_demo here>

<run newNAGsolver.xls>
Some routines available in Chapter e04

- e04ab: minimize a function of one variable
- e04dg: minimization using conjugate gradients
- e04mf: linear programming
- e04nc: linear least-squares
- e04nf: quadratic programming
- e04nq: LP or QP (for sparse problems)
- e04un: nonlinear least-squares
- e04vh: general sparse constrained nonlinear
- e04wd: general nonlinear all-purpose
- etc.
New optimization coming at next Mark

Currently many optimization routines in NAG, but these have all been for *local optimization*. No guarantee about which minimum (or maximum) is returned.
Local optimization

- Local Minimum
- Global Minimum
- Search start points

Possible paths
Global requirements

Users often ask for *global optimization* methods.

In next releases of NAG Libraries we will have software based on 'multilevel coordinate search' (MCS) method - Huyer and Neumaier:

http://www.mat.univie.ac.at/~neum/ms/mcs.pdf

*Search space is recursively split into sub-boxes, looking for child boxes where gain in objective is expected. Boxes swept through in turn, perhaps being split, until a box with maximum level exists. Then a local search is performed.*

Already in NAG Engine - new Chapter e05

Beta available now on request
New NAG Chapter – E05

- Main routine named E05JB
- Plus initialization and option setting routines
- Currently handles only bound constraints:

Minimize $f(x_1, x_2 \ldots x_n)$

Subject to $l_i \leq x_i \leq u_i$

<run e05jb_demo here>
Next release of the library imminent

- New global optimization chapter
- Nearest Correlation Matrix
- Partial Least Squares Regression Analysis
- Option Pricing
- Prediction intervals for fitted models
  - Allow for uncertainty in forecasts
- Fast quantile selection routine
- Wavelets
  - Data compression, edge detection
- Adoption of LAPACK 3.1
- New Random Number Generators
  - Including Mersenne Twister
  - Sobol Sequence generator (50,000 dimensions)
Use of NAG Software in Finance

- Portfolio analysis / Index tracking / Risk management
  - Optimisation, 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
- …..
NAG’s New Option Pricing Functions

- **Closed form solutions** with Greeks which provide a reference framework for approximate numerical methods: Monte Carlo, PDE, Trees
  - Written specifically for teaching* in collaboration with
    - Mike Giles
    - Ser-Huang Poon
    - William Shaw
    - Nick Webber
  - Available in C and Fortran with C++, Fortran and MATLAB interfaces

*there will be circumstances where the functions are useful for the real practitioner
Functions – set 1

- European options:
  - Black-Scholes-Merton
  - Lookback – Floating-Strike
  - Binary – Cash-or-Nothing
    Asset-or-Nothing
  - Barrier – Standard
  - Jump-diffusion – Merton Model
  - Heston’s Stochastic Volatility Model

- American options:
  - Bjerksund & Stensland (2002) approximation

- Asian options:
  - Geometric Continuous Average-Rate
Functions – set 2

- European:
  - Jump-Diffusion
    - Bates
  - Lookback
    - Fixed-strike
    - Partial Time – floating/fixed-strike
  - Barrier
    - Double
  - Stochastic Volatility
    - SABR

- European continued..:
  - Piecewise-Linear
    - Butterfly
    - Straddle
    - Condor

- Asian
  - Arithmetic

- American Options:
  - Barone-Adesi & Whaley
The Greeks – sensitivities to parameters

- Delta
  option price to underlying price
- Gamma
  delta to underlying
- Vega
  option price to volatility
- Theta
  option price to time to expiry
- Rho
  option price to risk-free interest rate
- Rhoq
  option price to dividend
- Vanna
  Delta to volatility
- Charm
  Delta to expiry
- Speed
  third derivative of option price to underlying
- Colour
  Gamma to time to expiry
- Zomma
  Gamma to volatility
- Vomma
  Vega to volatility
Option pricing – accessibility

- C
- C++
- C#
- Excel
  - Via Function Wizard
  - Via Menu
- FORTRAN
- MATLAB (via NAG Toolbox)

<run optionpricing_demo here>
A C++ example interface

europeanAnalytic BSEuro(PutCall, m, n, strike, spot, expiry, volatility, rate, dividend);

BlackScholesFormula calculateBSEuro(BSEuro);

calculateBSEuro.getPrice()
calculateBSEuro.getDelta()
Other NAG software

- **Fortran Builder** (NAG’s Windows Fortran compiler)
  <run Fortran Builder here>
- Maple-NAG Connector
- NAG’s High Performance libraries
  - SMP and Cluster parallelism
- Visualisation (IRIS Explorer…)

London Universities - 1st December 2008
NAG Fortran Builder

http://www.nag.co.uk/nagware/np/fortranbuilder.asp
Fortran Builder

- Integrated Development Environment for NAG compiler on PC
- Extra facilities: tools etc
- Excellent compiler for checking program validity.
- Implements many Fortran 2003 features
- Used extensively by NAG to test our library code
Summary

- Libraries of mathematical/statistical components for all your favourite environments:
  - FORTRAN, C, C++, C#, VB, Java, Python…
  - MATLAB, Maple, R,…

runs under all popular Operating Systems
  - Windows, Linux, Mac, Solaris,

- Other Environments:
  - Excel, Java, Python, R & C
CONTACT DETAILS

Technical Support  support@nag.co.uk

General:  operations@nag.co.uk

Presenter Contact Details:
john.holden@nag.co.uk
david.sayers@nag.co.uk
louise.mitchell@nag.co.uk

Copies of example programs used available on request
www.nag.co.uk

NAG Products http://www.nag.co.uk/products_and_services.asp
Downloads & evaluations/trials http://www.nag.co.uk/downloads/downloads_entry.asp

NAG C Library http://www.nag.co.uk/numeric/CL/CLdescription.asp
NAG Fortran Library http://www.nag.co.uk/numeric/fl/FLdescription.asp
NAG Toolbox for MATLAB http://www.nag.co.uk/numeric/MB/start.asp
Maple-NAG Connector http://www.nag.co.uk/numeric/MC/MCdescription.asp

NAG Fortran Builder http://www.nag.co.uk/nagware/np/fortranbuilder.asp

NAG and Excel http://www.nag.co.uk/numeric/callingDLLsfromotherlang.asp
NAG and Java http://www.nag.co.uk/doc/TechRep/html/Tr1_04/Tr1_04.asp
NAG and R http://www.nag.co.uk/numeric/RunderWindows.asp

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