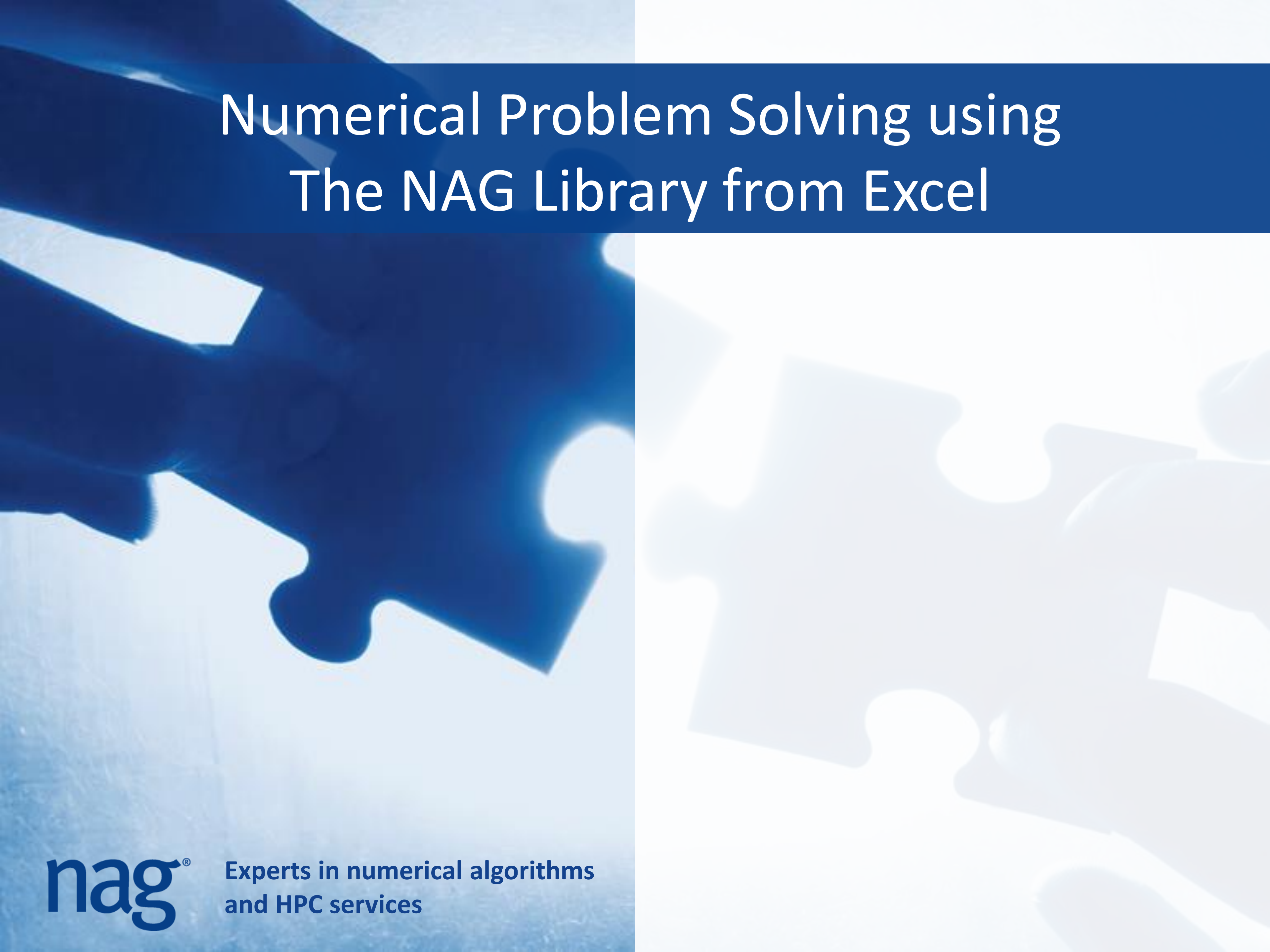

Numerical Problem Solving using The NAG Library from Excel

Experts in numerical algorithms
and HPC services

Program

- Overview of the NAG Library
- Quick Demonstration
- Some Worked Examples
- Practical

The NAG Library

- Can be thought of as:
  - Single code base
  - Multiple interfaces
- Interfaces:

Fortran	(NAG Fortran Library)
C	(NAG C Library)
.NET (C#)	(NAG .NET Library)
MATLAB	(NAG Toolbox for MATLAB)
Statistical Add-ins for Excel	

Show Installation Directory

What Do You Get?

- Where is it?

  C:\Program Files (x86)\NAG\FL25\fldll254ml

- A number of DLLs and lots of Fortran stuff
- Some Excel examples

  samples\excel_examples\

- VB6 headers

  vb6_headers\

- Documentation (possibly)
- Full documentation and additional examples from:
  - □ www.nag.co.uk

What Can I Do With it? – A Demo

Fitting a Variance Gamma distribution to data

PDF has 4 parameters (c, σ, θ and ν) and is given by:

$$f(x)=\frac{2\exp\left(\theta(x-c)/\sigma^2\right)}{\nu^{1/\nu}\Gamma(1/\nu)\sigma\sqrt{2\pi}}\left(\frac{|x-c|}{\sqrt{2\sigma^2/\nu+\theta^2}}\right)^{1/\nu-1/2}K_{1/\nu+1/2}\left(\frac{|x-c|\sqrt{2\sigma^2/\nu+\theta^2}}{\sigma^2}\right)$$

Moments:

$$E(X)=\mu=c+\theta$$

$$E((X-\mu)^2)=\sigma^2+\theta^2\nu$$

$$E((X-\mu)^3)=2\theta^3\nu^2+3\sigma^2\theta\nu$$

$$E((X-\mu)^4)=3\sigma^4\nu+12\sigma^2\theta^2\nu^2+6\theta^4\nu^3+3\sigma^4+6\sigma^2\theta^2\nu+3\theta^4\nu^2$$

Show Demo

NAG Library Contents

- **C05: Root Finding**
- C06: Summation of Series
- D01: Quadrature
- D02: ODEs
- D03: PDEs
- D04: Numerical Differentiation
- D05: Integral Equations
- E01: Interpolation
- E02: Curve and Surface Fitting
- **E04: Local Optimization**
- E05: Global Optimization
- F: Linear Algebra
- **G01: Statistical Functions**
- G02: Correlation / Regression
- G03: Multivariate Methods
- G05: RNGs
- G07: Univariate Estimation
- G08: Nonparametric Statistics
- **G10: Smoothing in Statistics**
- G12: Survival Analysis
- G13: Time Series Analysis
- H: Operations Research
- **S: Special Functions**
    - Option pricing

Show Documentation

NAG Documentation

- Organised in chapters, by functionality
  - Strange, but structured, naming
- Each chapter has an introduction
  - Overview of the problems
  - Suggested routines, often with a flow chart
- Each routine has an individual document
  - Routine prototype
  - Description and references
  - Description of arguments
  - Description of possible error exits

Programming for Excel

- VBA
  - Visual Basic for Applications
  - Comes as part of Excel
  - No compiler is required
  - Was going to become depreciated – but maybe not now
- COM
  - Usually C based, requires compiler
- VSTO
  - Relatively new
  - .NET based, requires compiler

Show Example 1

log Gamma function

$$\ln\left(\Gamma\left(x\right)\right)$$

Example 1: VBA – part 1

- VBA accessed via Developers Tab
  - Turned on in File -> Excel Options -> Customize Ribbon

- May need to alter security settings to allow macros to run
  - File -> Trust Center -> Trust Center Settings -> Macro Settings

Example 1: VBA – part 2

- Option Base 1
  - □ Not required
- Option Explicit
  - □ Doesn't always warn, especially if variable is an array
- Third party libraries accessed via Declare statement
  - □ **Declare Function** <*name*> **lib** <*location*> <*prototype*>
  - □ **Declare Sub** <*name*> **lib** <*location*> <*prototype*>
- Rename third party routines via "alias"
  - □ **Declare Sub** <*new name*> **lib** <*location*> **alias** <*old name*>

Example 1: NAG

- Declarations are supplied for all NAG routines
  - Use VB6 declarations
- Error handling via IFAIL
  - On entry: Three possible values, 0, -1 or 1.
    - IFAIL = 0 (noisy, hard return). **Don't use**, will close Excel.
    - IFAIL = -1 (noisy, soft return). Uses a (non-Excel) pop up window.
    - IFAIL = 1 (quiet, soft return). Recommended
  - On exit:
    - IFAIL = 0 means everything is OK
    - IFAIL <> 0 is either a warning or error
    - Returned value is a numeric code which can be looked up
    - Good practice to test for non-zero values

Show Example 2

Summary Statistics

Example 2: VBA

- Array Functions
  - Allow a function to return more than one value
  - Dynamic
  - Usually return a 2D variant array
    - Can return a 1D variant array, but will be a row vector
    - Can be an array of a different type (i.e. double), but then can’t be used to return error messages etc
  - Expanded using Shift+Ctrl and Return
  - Extra space is filled with #N/A
  - Access individual elements of array via functions like INDEX, i.e. “= index(myFun(),4,1)” returns element (4,1)
    - Note: Multiple uses of index causes myFun to run multiple times

Example 2: NAG

- All arguments are passed by reference
- Must supply the first element of an array, rather than the array itself, so:

  ```
  CALL G01AAF(N, X(1), ...)
  ```

  rather than

  ```
  CALL G01AAF(N,X, ...)
  ```

- Same applies to 2D arrays:

  ```
  CALL G02AAF(G(1,1), ...)
  ```

- 2D arrays are stored in column major order

Show Example 3

Modified Bessel Function

$$K_v(x)$$

Example 3: VBA

- Types can be defined using:

```
Type Complex
      Real_Part as Double
      Complex_Part as Double
 End Type
```

- Types can be accessed using:

```
Dim z as Complex
z.Real_Part = 0.0
z.Complex_Part = 1.0
```

Example 3: NAG

- Any user defined types required by a NAG routine are supplied in the declaration file

- Routines with character (or string) arguments have a “hidden” argument:
  - □ corresponds to the length of the character argument
  - □ will not appear in the documentation
  - □ does appear in the VB declaration
  - □ string arguments and their lengths are one of the few passed by value

Show Example 4

System of Non-Linear Equations

$$\begin{aligned}
(3-2x_1)x_1 - 2x_2 &= -1 \\
-x_{i-1} + (3-2x_i)x_i - 2x_{i+1} &= -1 \quad i = 2,3,\ldots,8 \\
-x_8 + (3-2x_9)x_9 &= -1
\end{aligned}$$

Example 4: VBA

- **RtlMoveMemory** can be used to copy memory
  - □ **RtlMoveMemory**(*to*,*from*,*amount*)
  - □ in kernel32 DLL
  - □ comes with windows
- **AddressOf** can be used to access the address of a function or subroutine
- Len function can be used to obtain the size of a type

```
Dim y(10) as double, x(10) as double, size_double as long
size_double = Len(x(1))
Call RtlMoveMemory(x(1), y(1), n*size_double)
```

Example 4: NAG

- User callable functions (or subroutine) must be passed to NAG routines using **AddressOf**
- IUSER and RUSER can be used to pass information
- In User callable routines:
  - Arrays in the routine argument list are pointers (long's)
  - Data must be copied out of and in to these arrays using **RtlMoveMemory**
  - Most NAG examples alias **RtlMoveMemory** to **CopyMemFromPtr** and **CopyMemToPtr**. Which have different **ByVal** / **ByRef** pattern.
- Declaration file has prototype for callable functions

Where Can I Get Additional Help?

- Examples supplied with library
- NAG website
  - □ http://www.nag.co.uk/numeric/nagandexcel.asp
- Mail support
  - □ mailto:support@nag.co.uk

Practical