Software Support for Metrology Best Practice Guide No. 5

Software Re-use: Guide to METROS

Robin Barker

September 2000

Software Support for Metrology Best Practice Guide No. 5

Software Re-use: Guide to METROS

Robin Barker
Centre for Mathematics and Scientific Computing

September 2000

ABSTRACT

The guide describes METROS (the METROlogy Software environment) which is a system to provide metrologists access to software appropriate to their needs. It gives an overview of the system and describes the conceptual structure of METROS. The user will access METROS via the WWW and we describe the externally visible structure of the web interface. We explain the internal structure of METROS and how the various components should be kept in alignment. We also explain what is required to make contributions to METROS and how new components are added.

ISSN 1471–4124

Authorised by Dr Dave Rayner,
Head of the Centre for Mathematics and Scientific Computing

National Physical Laboratory,
Queens Road, Teddington, Middlesex, United Kingdom TW11 0LW

Contents

List of Figures

1 Introduction

This best practice guide was produced by the Software Support for Metrology (SS*f*M) programme as part of the *Software Re-use* Project; it provides a metrology software library best practice guide. The *Software Re-use* Project was led by the Numerical Algorithms Group (NAG Ltd.), collaborating with the National Physical Laboratory.

The guide describes METROS (the METROlogy Software environment) which is a system to provide metrologists access to software appropriate to their needs. METROS is a state of the art solution to the provision of a metrology software library and is expected to set the model to be followed in the future.

The first part of the guide is aimed at the user of METROS: it gives an overview of the system and describes the conceptual structure of METROS. The user will access METROS via the WWW and we describe the externally visible structure of the web interface. Then we show, by example, how the user is likely to access and interact with METROS.

The later sections of the guide are aimed at those who will maintain and enhance METROS. We explain the internal structure of METROS and how the various components should be kept in alignment. We also explain what is required to make contributions to METROS and how new components are to be added.

The SS*f*M programme is managed on behalf of the DTI by the National Physical Laboratory. For more information on the SS*f*M programme visit the web site at www.nplco.uk/ssfm, or contact the programme manager Dr Dave Rayner on +44 (0) 20 8943 7100 (e-mail: ssfm@npl.co.uk) or by post to National Physical Laboratory, Teddington, Middlesex, UK TW11 0LW.

2 Overview of METROS

2.1 Aims

The METROS system allows metrologists to gain access to the software they require for their work. The central idea is that of the *key function*. This is either a function that may be used in many different applications or a function that is seen as being fundamental to an area in metrology. A list of the key functions to be included in METROS was developed by the *Software Re-use* project and is given in annex A. This list intends to cover the fundamental computational requirements of metrologists. These functions are described in METROS by their mathematical specification.

Most metrologists may not be familiar with these functions from a mathematical perspective but will view their own problems from an application perspective. METROS therefore contains application functions based on key functions and links between these application functions and the key functions.

Further, metrologists will be working in different computing environments and platforms. They will require the appropriate software for their platforms. Finally, they will benefit from the support of appropriate case studies and best practice guides which will also be contained in METROS.

The aim of METROS is to provide good re-usable software. This software may be free or it may be provided by commercial software producers. In many cases the user will have the choice between a free implementation of a function or a commercial implementation; the user will have to decide whether the greater support available with the commercial implementation is worth paying for.

2.2 Components of METROS

The major components of METROS are the metrology areas, the key functions, the application functions, the implementations of these functions and the supporting case studies and best practice guides. Associated with each component is the information about the component and links to the components which are associated with a given component. For instance: a metrology area component will have information about the metrology areas and a list of application functions for that area; a key function component will have the elements that describe the functionality of the key function and a list of implementations of the function.

Some functions used by metrologists are implemented in different ways because different mathematics is used to specify the same (natural language) requirements. Also, METROS needs to be able to deal with functions which address some real problem for which no adequate specification has been produced. Another component has been introduced to allow METROS to describe these functions, called a generic key function. A generic key function has some of the elements of a key function but it does not have a (mathematical) computational aim, only a "purpose" expressed in natural language. Generic key function components are used to collect together key functions which are specifications of the same purpose and to record requirements for functions with a given purpose for which no key function specification is available.

The relationship between the function components of METROS, from most general and least specified down to most specific and fully specified, is shown in figure 1. The best practice guides are not shown in this figure because they do not relate to a particular function. Indeed, there will be further components of METROS to cover best practice guides, software packages and libraries, and testing methods.

This guide to the use of METROS will be expanded to include these other components when they are added.

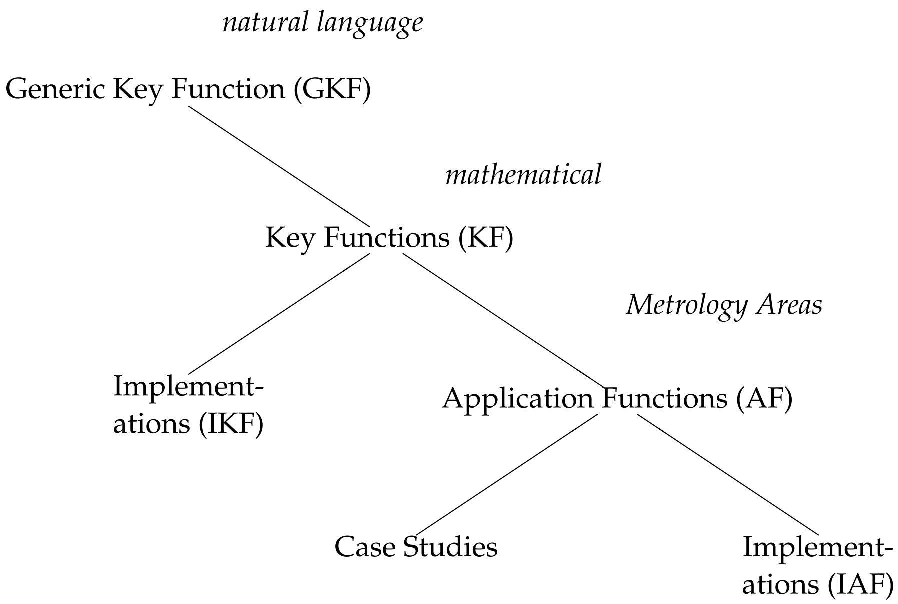


Figure 1: METROS function components

2.3 METROS community

METROS is a collaborative venture with different sections of the metrological and software industries contributing to and using the system. The METROS community should be seen as involving four different groups:

(a) End-users

(b) Contributors

(c) Library and system providers

(d) Co-ordinators

In practice, individuals and organisations will be represented in more than one group.

(a) End-users

An end-user will be anyone who uses the system to provide them with their required software. They will fall broadly into two classes: metrologist end-users who need the software for their own use or to be used by their customers; and computer programmers contracted to metrologists who have been given the task to create the required software. The main differences between the two groups are the skills and knowledge they have. Those with sufficient programming and application system skills will be able to adapt functions available in one language/system to another. Those with sufficient mathematical knowledge will naturally relate their problems to the key functions.

(b) Contributors

Contributors will provide components to be included in the METROS system. These may be key functions, applications, interfaces to specific systems, best practice guides etc.

Contributors will be experts in their areas. They may wish to contribute to METROS for a variety of motivations depending on their circumstances. For those working in academic and government organisations it would be part of the dissemination of their work. For those from commercial organisations it could be to gain publicity and promote the use of their software.

(c) Library and system providers

Much of the underlying software required by the key functions is likely to be from Library and system providers. The providers may be commercial or non-commercial (public domain software). In the case of commercial suppliers, the users would have to pay a licence fee for the use of the software but would have the possibility of support. The providers of public domain software would not charge a fee but also would not provide any support. It is likely that the use of public domain software will come through contributions to METROS as described above.

For a commercial software house the METROS system provides an opportunity for them to display their software to a specialist audience. Their requirements will be that it is suitably referenced and that there is clear information on licencing and contact with the organisation.

(d) Co-ordinators

The co-ordinators of METROS are providing a service to the metrology community. Their aims will be to increase the awareness of the tools that are available and how they should be used. They will aim for the maximum involvement in METROS by contributors and users. Thus, their main requirement is that the needs of these groups are met. In addition, they will require METROS to run without consuming too much resource. Therefore, there is a need for a robust structure that is easy to maintain and a clear, straightforward procedure for inclusion of contributions in the system.

2.4 Summary

The METROS system has components to enable the re-use of metrology software. These components and the infrastructure behind them meet the essential requirements of the METROS community; the requirements can be summarised as:

1. Ease of use for end-users
2. Good coverage and reliability in providing the correct software
3. Clear procedures for inclusion of contributions
4. A robust system that is easy to update
5. Links to software providers

3 Web view of METROS

3.1 METROS web site

The METROS web site (see figure 2) is primarily a means of accessing information about functions which form the Metrology Software library. It will also be possible to access software which implements the key functions, in various formats, as well as further written material such as case studies and best practice guides.

There are different web pages for the different components, the information for each component being provided in a tabular form. The tables provide a description of the component and its elements, and includes lists of related components. The main entry points into METROS will be the metrology areas and the key functions, but it is also possible to start from (for example) the generic key functions.

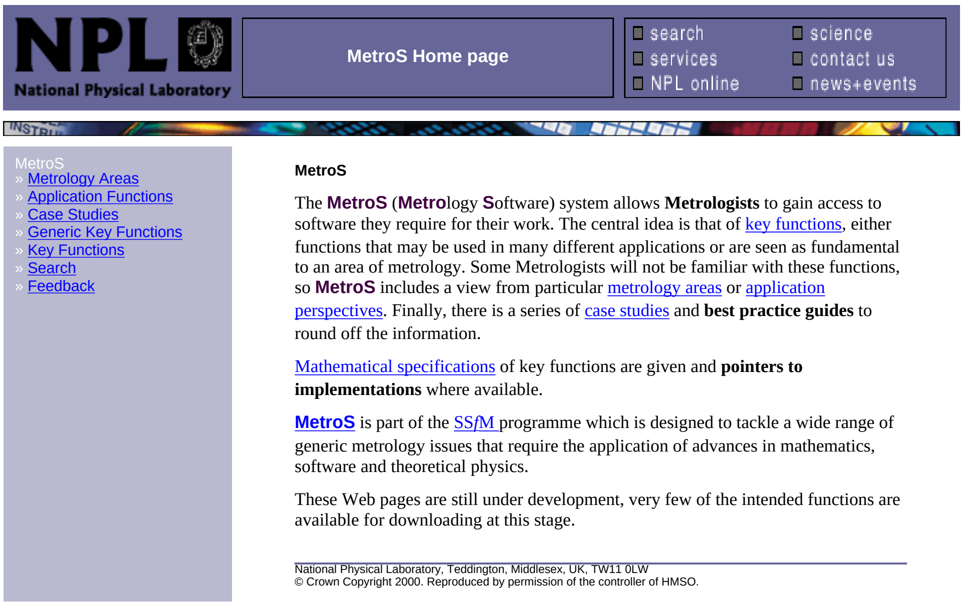


Figure 2: METROS home page

3.2 Component web pages

3.2.1 Metrology Areas (MA)

This will be the metrologists' usual entry point into METROS. Metrology areas follow the existing NMS classification, see figure 3.

On each individual metrology area page there is a list of the application functions, a list of relevant generic key functions, and a list of case study material for that area.

In the case of SS*f*M, its page will list the best practice guides and other useful generic guidance material. Other NMS programmes which relate equally to all metrology areas (e.g. the National Measurement Partnership programme) are not included. The NMS programmes are changing: new programmes being added and old programmes being renamed; this list of Metrology Areas will need to updated in the future to reflect these changes.

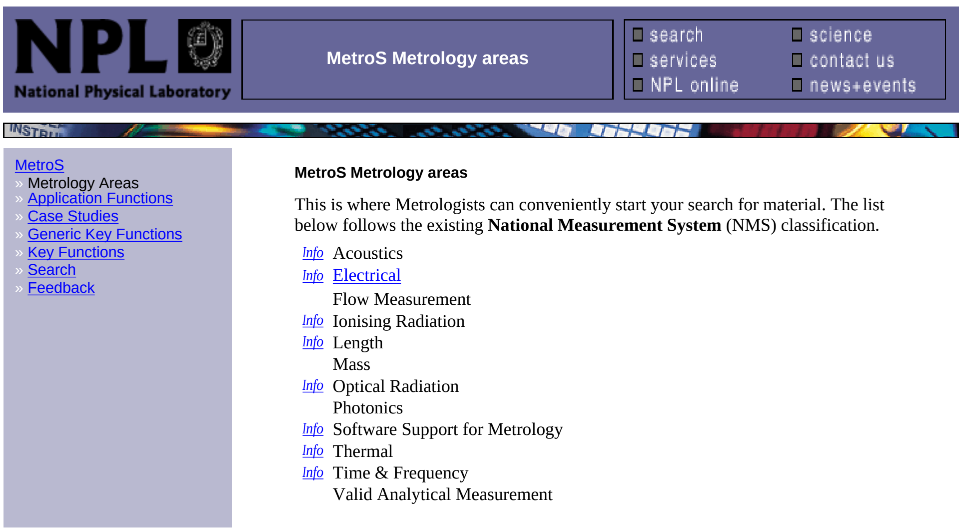


Figure 3: Metrology areas page

3.2.2 Generic Key Functions (GKF)

Generic key functions are the simplest functional components of METROS. A generic key function is "a problem without a solution": it defines the purpose of a computation without specifying a mathematical technique for finding a solution. They are listed according to subject, in the same way as key function (see section 3.2.3).

The web page (see figure 4) gives a natural language description of a function, with some of the parameters that such a function would use.

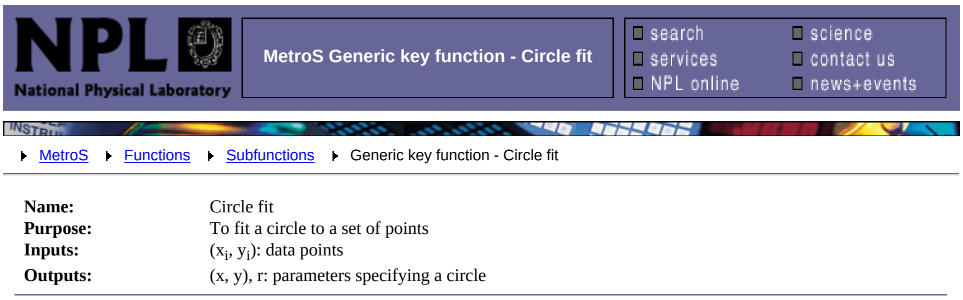


Figure 4: Generic key function page

3.2.3 Key Functions (KF)

These are the central function components of METROS. They are listed according to category (e.g. geometry, statistics, fitting routines, properties of materials).

There is a main key function web page and a web page listing the key functions in each category, as shown in figure 5 and figure 6

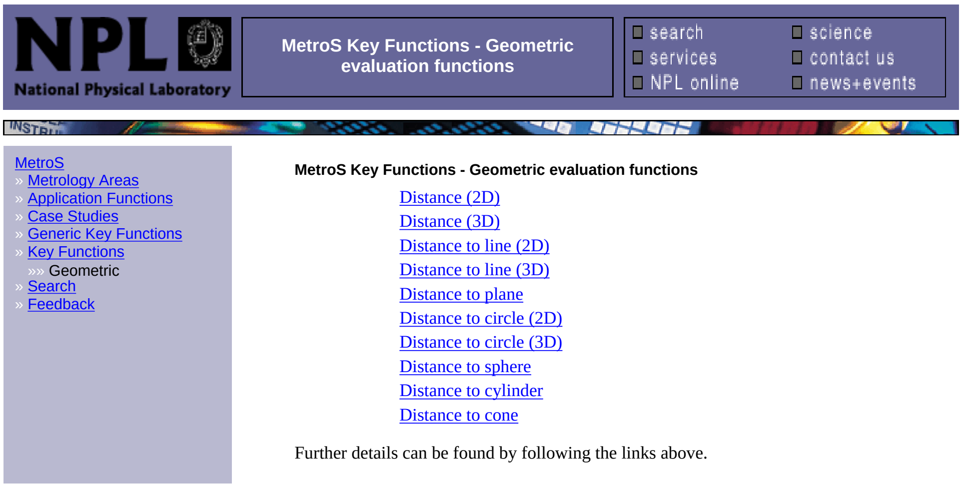


Figure 5: Key function category page

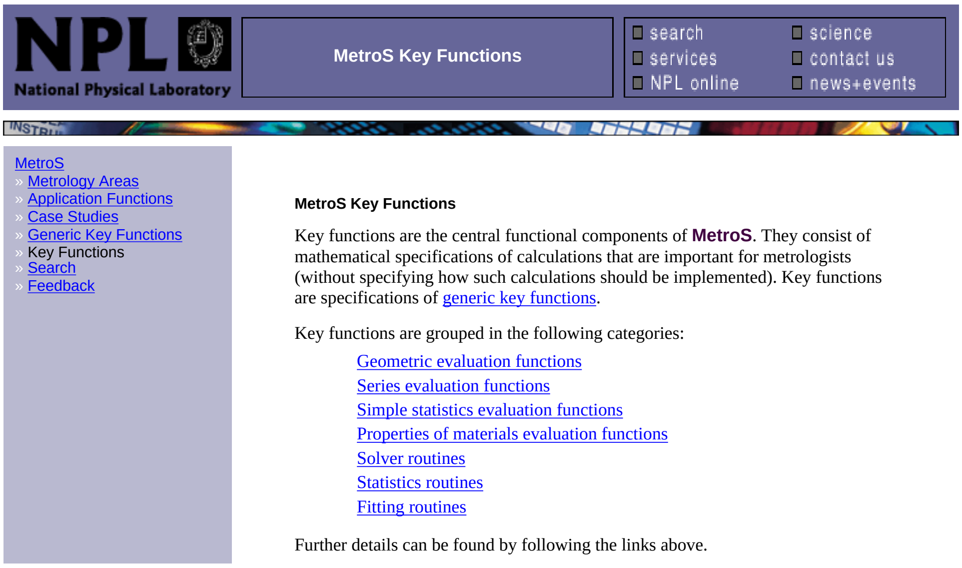


Figure 6: Key function list page

The web page for a key function (see figure 7) gives details specifying the function: purpose, method, inputs and outputs, computational aim and how implementation is to be validated. It also gives the corresponding generic key function (if any) and lists the implementations of this key function.

NPL National Physical Laboratory | MetroS Key function Distance to circle (2D) | search | science | services | contact us | NPL online | news+events

MetroS ▸ Functions ▸ Subfunctions ▸ Distance to circle (2D)

Name: Distance to circle (2D)
Purpose: To calculate the distance of a point from a circle in 2D
Method: geometric elements
Generic:
Inputs:

Name	Type	Description
(x, y)	Double: pair	point
(x_0, y_0), r	Double: pair, scalar	parameterisation of a circle

Outputs:

Name	Type	Description
d	Double	distance
J	Double: array (3)	(OPTIONAL) Jacobian

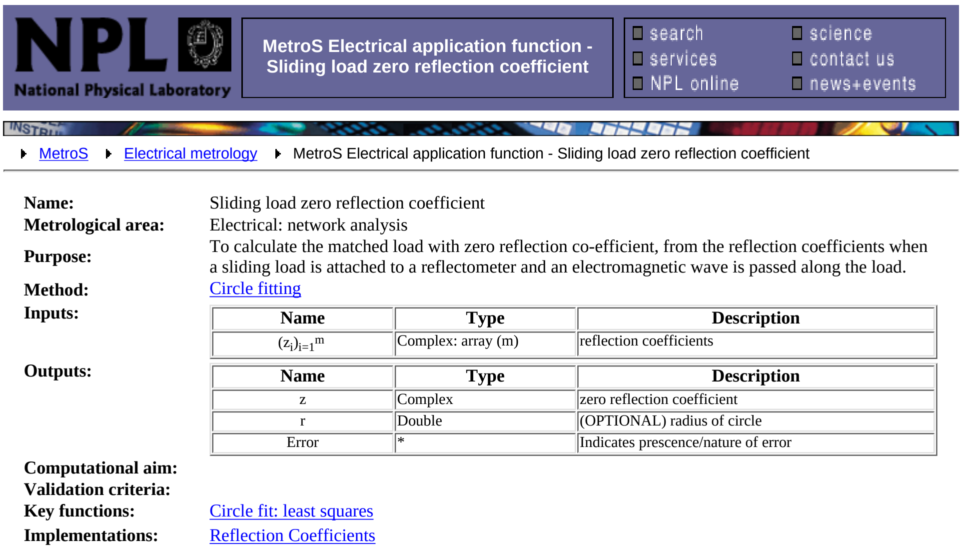
Computational Aim: d = min $_{p \text{ in } C}$|| (x,y) - p ||, where C is the circle {p: ||p - (x_0,y_0)|| = r}
J is the Jacobian of d w.r.t. x_0, y_0, r
Validation Criteria: reference data sets, geometric characterisation
Implementations: NAG Fortran source
NAG Fortran dll
NAG Excel: available as part of the NAG Excel implementation of circle-fit: least-squares

Figure 7: Key function page

3.2.4 Application Functions (AF)

These are the functional components of METROS that a user is most likely to use directly, consisting of a key function (or combination of key functions) adapted to a particular metrology area.

The web page for an individual application function has a similar structure to a key function web page, see figure 8.



NPL National Physical Laboratory

MetroS Electrical application function - Sliding load zero reflection coefficient

search | science | services | contact us | NPL online | news+events

MetroS ▸ Electrical metrology ▸ MetroS Electrical application function - Sliding load zero reflection coefficient

Name: Sliding load zero reflection coefficient

Metrological area: Electrical: network analysis

Purpose: To calculate the matched load with zero reflection co-efficient, from the reflection coefficients when a sliding load is attached to a reflectometer and an electromagnetic wave is passed along the load.

Method: Circle fitting

Inputs:

Name	Type	Description
$(z_i)_{i=1}^{m}$	Complex: array (m)	reflection coefficients

Outputs:

Name	Type	Description
z	Complex	zero reflection coefficient
r	Double	(OPTIONAL) radius of circle
Error	*	Indicates prescence/nature of error

Computational aim:

Validation criteria:

Key functions: Circle fit: least squares

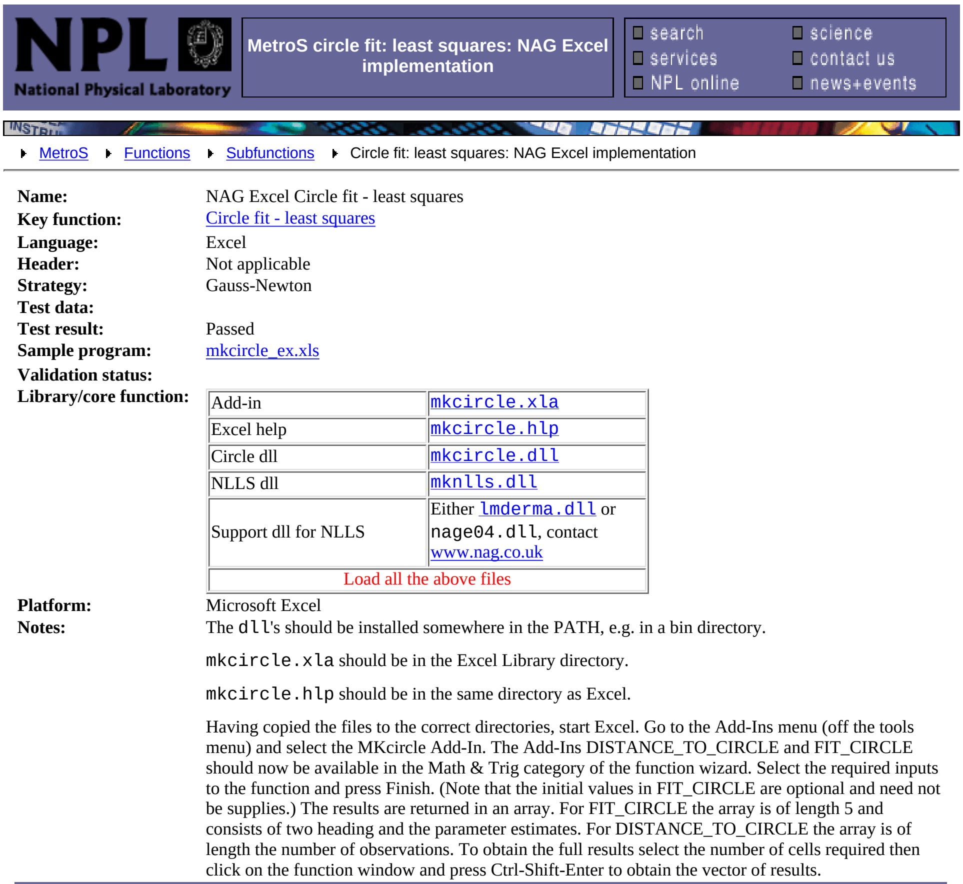
Implementations: Reflection Coefficients

Figure 8: Application function page

3.2.5 Implementations of Key Functions (IKF)

The key function implementation components of METROS contain information on available implementations of key functions. These include both source code implementations and executable code implementation for a particular platform (computing environment).

The page for a key function implementation contains documentation for the function, including additional parameters (beyond those specified for the key function), information on accuracy and valid parameter ranges, and the meaning of error codes. The implementation may be immediately available from METROS or there may be a link to the software supplier. Figure 9 shows an Excel implementation and figure 10 shows a Fortran implementation.


NPL National Physical Laboratory

MetroS circle fit: least squares: NAG Excel implementation

search
science
services
contact us
NPL online
news+events

MetroS ▸ Functions ▸ Subfunctions ▸ Circle fit: least squares: NAG Excel implementation

Name: NAG Excel Circle fit - least squares
Key function: Circle fit - least squares
Language: Excel
Header: Not applicable
Strategy: Gauss-Newton
Test data:
Test result: Passed
Sample program: mkcircle_ex.xls
Validation status:
Library/core function:

Add-in	`mkcircle.xla`
Excel help	`mkcircle.hlp`
Circle dll	`mkcircle.dll`
NLLS dll	`mknlls.dll`
Support dll for NLLS	Either `lmderma.dll` or `nage04.dll`, contact www.nag.co.uk
Load all the above files	

Platform: Microsoft Excel
Notes: The `dll`'s should be installed somewhere in the PATH, e.g. in a bin directory.

`mkcircle.xla` should be in the Excel Library directory.

`mkcircle.hlp` should be in the same directory as Excel.

Having copied the files to the correct directories, start Excel. Go to the Add-Ins menu (off the tools menu) and select the MKcircle Add-In. The Add-Ins DISTANCE_TO_CIRCLE and FIT_CIRCLE should now be available in the Math & Trig category of the function wizard. Select the required inputs to the function and press Finish. (Note that the initial values in FIT_CIRCLE are optional and need not be supplies.) The results are returned in an array. For FIT_CIRCLE the array is of length 5 and consists of two heading and the parameter estimates. For DISTANCE_TO_CIRCLE the array is of length the number of observations. To obtain the full results select the number of cells required then click on the function window and press Ctrl-Shift-Enter to obtain the vector of results.

Figure 9: Excel key function implementation page

NPL National Physical Laboratory

MetroS circle fit: least squares: NAG Fortran implementation

search | science
services | contact us
NPL online | news+events

▸ MetroS ▸ Functions ▸ Subfunctions ▸ Circle fit: least squares: NAG Fortran implementation

Name: NAG Fortran source Circle fit - least squares
Key function: Circle fit - least squares
Language: Fortran 77
Header: Not available
Strategy: Gauss-Newton
Test data:
Test result: Passed
Sample program:
Validation status:
Library/core function:

Fortran source	`mkfcir.f`
Load all the above files	

Platform:
Notes:

Parameters:

SUBROUTINE MKFCIR(X,Y,N,INW,W,X0,Y0,R,IERROR)		
Scalar Arguments	DOUBLE PRECISION R,X0,Y0 INTEGER IERROR,INW,N	
Array Arguments	DOUBLE PRECISION W(*),X(N),Y(N)	
X(N)	INPUT	X coordinates of points.
Y(N)	INPUT	Y coordiantes of points.
N	INPUT	Number of points (Implementation specific), N > 0.
INW	INPUT	If INW > 0 wieights are used (Implementation specific) .
W(*)	INPUT	The weights, only referenced if requested by INW, in which case W must be of dimension at least 3 and W(i) >= 0.
X0	INPUT/OUTPUT	Initial/final estimate of X coordinate of centre of circle.
Y0	INPUT/OUTPUT	Initial/final estimate of Y coordinate of centre of circle.
R	INPUT/OUTPUT	Initial/final estimate of radius of circle, R > 0.
IERROR	OUTPUT	Indicates errors in parameters: IERROR = -3: N <= 0 IERROR = -5: W(i) < 0 IERROR = -8: R <= 0 IERROR = 1: Not converged IERROR > 1: doubts about the solution increasing with value IERROR = 8: should not occur

Algorithm

Uses METROS key functions "non-linear least-squares" to compute the parameters. NOTE: it depends on the implementation of "non-linear least-squares" having the required specification for the objective function. Residuals and derivatives are computed using METROS key function "Distance to circle"

Accuracy

Depends on the implementation of METROS key function "non-linear least-squares"

Libraries

As required by the implementation of METROS key function "non-linear least-squares"

Figure 10: Fortran key function implementation page

3.2.6 Implementations of Application Functions (IAF)

The application function implementation components of METROS contain information on available implementations of application functions. They will be the useful functions for metrologists in a particular metrology area, whether or not that function has been identified as an existing function within METROS.

The web page for an application function implementation will be similar in structure to a key function implementation web page, see figure 11.

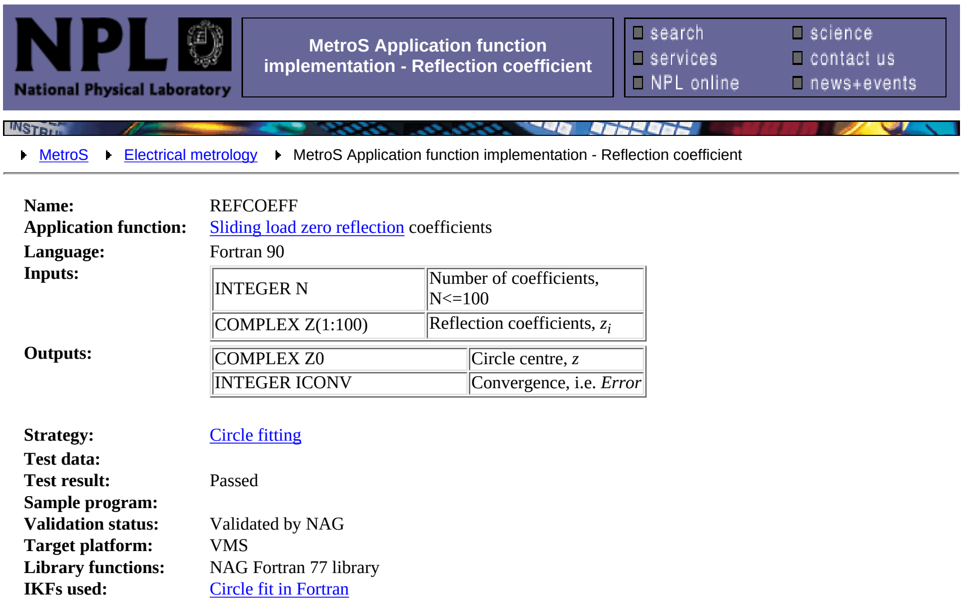


Figure 11: Fortran application function implementation page

3.2.7 Case Studies (CS)

Case studies of the use of particular functions from METROS will be an important way of showing metrologist end-users how functions can be used and re-used. Case studies will take the form of technical reports from the metrology community (with suitable editing from the METROS co-ordinators, if required). The web page will contain an abstract for the case study and a link to an on-line copy of the report or details of how to obtain a (printed) copy of the report.

3.3 Other web pages

The METROS web site contains pages for searching the whole site: this will allow the user a way to find information about a particular subject or keyword without knowing where it fits within the METROS structure.

The web site also allows for users to feedback comments to the METROS co-ordinators or to post requests or queries to a "chat" page (currently part of the SS*f*M web site).

In future, there will be pages for best practices guides from the SS*f*M programme, pages for various software packages and libraries, and pages for testing methods to be used to validate METROS functions.

4 Use of METROS

Users of METROS will have differing knowledge and skills and this will affect the way in which they interact with the system. For end-users with a programming/mathematical perspective, METROS will work as a repository of available functions; for these users the classification of key functions (or the generic key functions) by subject will be a suitable entry point. However, most users will have a metrological perspective and the system must provide a entry point and route to the required functions for those who do not wish to use the functionality oriented perspective.

For these metrologist-users, the entry point where they have most knowledge is likely to be the metrology area. The metrology area page provides links to application functions and on to (generic) key functions, so the metrologist-users will be led to the functions they need. The typical metrologist's access to METROS follows the steps illustrated in figure 12, starting from the metrology areas.

At each stage, the user will either find the functions the user is looking for or will be able to follow links to more general functions. If the user finds there is no application function in their metrology area, there may be a key function which provides the same functionality in a general form. The user can take an implementation of the key function and use it to create the required application function. If there is no implementation of the key function then the user can request the METROS community to produce an implementation, or give the user help as to how to produce their own implementation.

Alternatively, the user may look in the generic key function for a function which meets their needs. This may lead the user to one or more key functions which can be used (as above), or the user may again request that such a key function be specified and implemented.

If there is nothing which meets their needs, the user can request the inclusion of a generic key function.

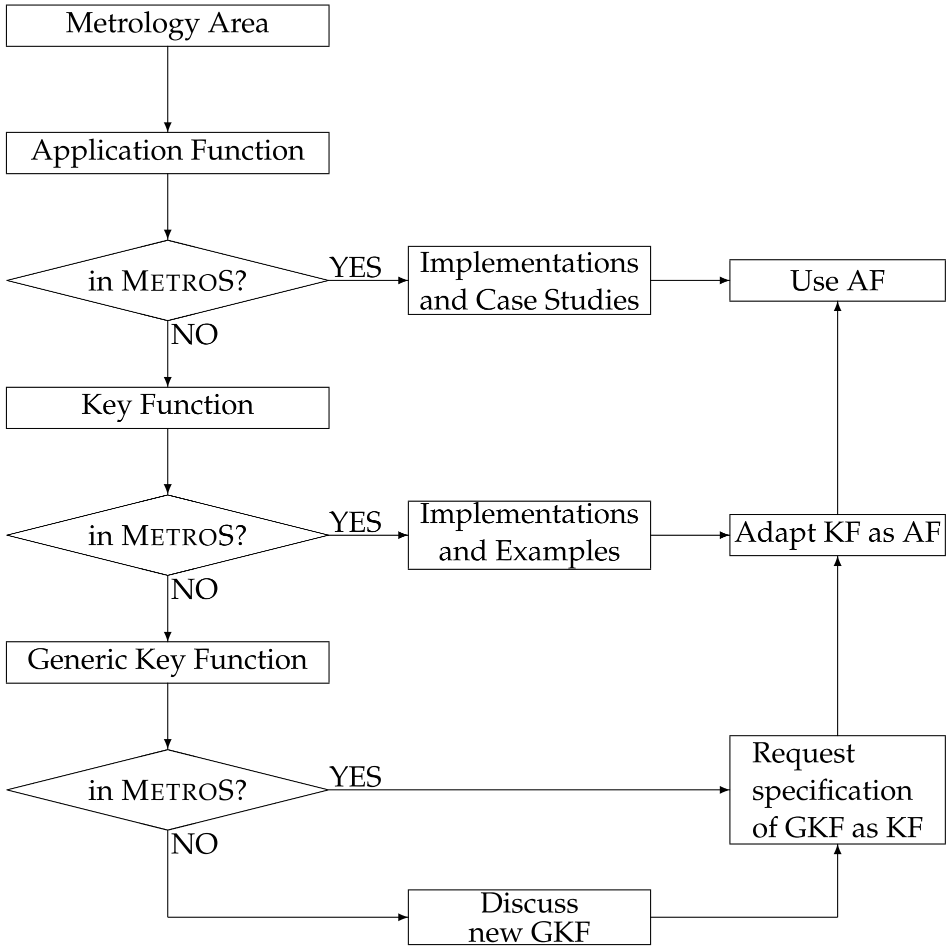


Figure 12: Metrologist's use of METROS

5 Internal structure of METROS

5.1 METROS components and fields

In abstract, METROS is an information access system which we shall think of as a relational database. At the moment this simply gives us a convenient language for speaking about the METROS design. METROS is currently realised as a number of manually generated web pages but as the system grows it will become necessary to automatically generate the web pages from some information access system. The web pages form the interface between the METROS system and the user via the Internet.

Each component of METROS will be an entry in the database, and the different types of component will have different fields. These fields indicate which information should be directly associated with the component; other information will be found in related entries. The web pages for a component use some of the information from the fields and some of the information on related entries.

5.2 Structure of each METROS component

Here we list the fields for each of the METROS component types and the types of component they are related to.

5.2.1 Metrology area

This component type has one field which identifies the NMS Metrology Area:

Acoustics,	Electrical,
Flow Measurement,	Ionising Radiation,
Length,	Mass,
Photonics,	Optical Radiation,
Thermal,	Time and Frequency,
Valid Analytical Measurement.	

In addition, SS*f*M will be listed on the Metrology Area web page to provide access to best practice guides and other useful generic guidance material. If required, the metrological area could be linked to other subject classification schemes, e.g., American Institute of Physics, American Mathematical Society.

The metrology areas are related to application functions, relevant generic key functions, and case studies from the area.

5.2.2 Generic key function

Fields:

- Name
- Purpose — problem being solved
- Inputs — names, types and meaning (Optional)
- Outputs — names, types and meaning (Optional)

Generic key functions (KF) are related to the key functions which are specifications of the functions and metrology areas in which the functions are used.

5.2.3 Key function

Fields:

- Name
- Purpose — problem being solved
- Method — name of mathematical technique to be used to implement the computational aim
- Corresponding generic key function
- Inputs — names, types and meaning
- Output — names, types and meaning
- Computational aim: relationship between i/o and constraints on i/o; and documentation.
- Validation criteria

  The validation criteria will include details of the test method and test data, such as reference data sets.

Key functions are related to generic key functions which they specify, and to application functions that use them.

5.2.4 Application function

Fields:

- Name
- Metrological area
- Purpose
- Method
- Inputs
- Outputs
- Computational aim:
  relationship between input/output (i/o) and constraints on i/o.
- List of KF which may be used to implement the function
- Validation criteria

Application functions are related to the metrology area components and to the key functions that they use.

5.2.5 Implementation of key function

Fields:

- Name
- Key function
- Language
- Signature
  - inputs: concrete types for KF inputs and any additional inputs
  - outputs: concrete types for KF outputs and any additional outputs
  - meaning/relationship/constraints on additional parameters

  The documentation of the additional parameters will include parameters used for exception and error handling.

- Strategy/algorithm/high-level design
- Test data
- Test results
- Sample program
- Validation status
- Library/core functions used
- Target platform or computing environment
- Availability: details of how to obtain the software; including source authentication, licensing arrangements, etc.

The Implementation of Key Function (IKF) component is related to the key function component and to the sources of the actual implementation code (source or executable).

5.2.6 Implementation of application function

An Implementation of Application Function (IAF) may be an existing function in use in metrology or it may be derived from an IKF following the relationship between the corresponding application function and key function.

Fields:

- Name
- Application function
- Language
- Signature (as for IKF)
- Strategy/algorithm/high-level design
- Test data
- Test results
- Sample program
- Validation status
- Target platform or computing environment

- Library/core functions used
- IKFs used
- Availability

5.2.7 Case study

Fields:

- Metrology area
- Who
- Where
- Best practice guides used or contributed to
- Platform
- Language(s)
- Key function(s)
- Application functions, implementations, executables.
- References/bibliography
- Abstract

Case studies are related to all other components that the report references.

6 Maintenance of METROS

Over time, METROS will grow, and the structure will evolve: we therefore need rules for how components are added.

There are different rules depending on the component.

1. New generic key functions could be proposed and discussed by anyone.

2. Key functions and application functions can only be added and changed by the parties who are co-ordinating METROS.

3. For implementations, these can be added freely (subject to some commercial considerations) but their status will be reflected in some "status" field, for example, untested, tested and validated.

4. Examples and case studies will be encouraged by the co-ordinators. They will be added after having been subject to some technical review (perhaps, by the co-ordinators themselves). Other METROS users can contribute comments on the usefulness of examples and case studies: these comments will be attached to the examples.

7 Contributions to METROS

7.1 Policy on contributions

The main requirement of contributors will be the availability of clear specifications for the inclusion of contributions in METROS. They may also like flexibility so that what they have already can be included in METROS with minimal change. This may lead to conflict between the need for clear specification and the needs of the co-ordinators of METROS on the one hand and contributors' desire for flexibility on the other. Therefore, there will be a requirement for a clear adjudication scheme with a final decision in each case.

Two types of contributions are expected: *validated* and *unchecked*. In the case of validated contributions, the contribution will have to be checked by the co-ordinators of METROS. For unchecked contributions, the status could be *untested* or *tested*, depending on the testing done by the contributor; there would only be minimal checks by the co-ordinators.

Commercial contributions to METROS are encouraged, and should provide publicity and sales for commercial software producers. Commercial software will probably be available in one of two forms; either the whole implementation will be available from the supplier (the web site will provide details), or there will be an implementation available on METROS but it will require use of commercial software libraries to execute. In the former case, the user will have to pay for the software; in the latter case, the user will need to have, or pay for, the software library. METROS will be extended to include information on these software libraries, and other software packages, themselves.

It is expected that METROS will not consist only of functions which fit neatly in the structure: some executable application function implementations may be built directly from library/core functions with no real use of (say) platform-independent key function implementations. However, the structure can cope with many different sorts of components. The way forward is to make the different components visible, so that other METROS users can develop (and generalise) a particular component to other platforms and application areas.

Contributions of guidance and testing material will also be encouraged, and will be used to extend METROS. The SS*f*M programme is producing much guidance in the form of best practice guides and these should be accessible from METROS. SS*f*M is also producing reports on software testing and producing test data (including reference data sets); these will be made available so they can be used to test and validate the software in METROS.

Contributions to METROS are welcomed from the international metrology community. The concept of a library of re-usable measurement software was proposed to EUROMET, and their interdisciplinary committee (INTMET) agreed to a project (EUROMET project 567) on "EUROMETROS", to add a European dimension to METROS. Fourteen countries on INTMET expressed support for the project, which will encourage contributions from other European NMIs. The METROS concept has been presented at international conferences and workshops, which will encourage further international contributions.

7.2 Technical issues

7.2.1 Use of XML templates

To aid the submission of key function specifications, there is an XML template for the key functions. This allows the basic content of the key function specification to be saved in a structured file which can then be processed to produce the HTML pages for the web site, including the standard headers, navigation buttons, and links to other pages. There are many editors which can work with XML, so the template should be easy to complete. An example of a completed template, for the key function "Polynomial Fit", is shown in figure 13. These templates were used to produce web pages for all the key functions listed in annex A.

7.2.2 Use of mathematics in contributions

Two ideas have been important to METROS since its inception: the use of the WWW as a delivery mechanism and the use of mathematical specifications to provide a rigorous description of key functions. This has lead to an issue that has not been properly solved: how to display mathematics on web pages. The mathematical content of contributions to METROS must be in a form that can be converted in to the contents of web pages.

The current solution is for contributions of key functions to use LaTeX to describe the mathematics. This can be seen in figure 13: mathematics to appear with text is marked by $s, and mathematics to be displayed on its own is marked by `\[` and `\]`. The software to produce the web pages from the templates includes a tool to convert LaTeX to HTML. This tool uses special symbols, superscripts, subscripts and tables in HTML to reproduce the mathematics. The quality of the resulting web pages varies from one piece of mathematics to another, and from one browser to another; but the web pages can be viewed with all common browsers.

```
<KF>
<Name>          polynomial fit
</Name>
<Purpose>       to fit a polynomial to data
</Purpose>
<Method>        least-squares
</Method>
<Inputs>
<Name>          $(x_i,y_i)_{i=1}^m$
</Name>
<Type>          Double: array ($m$) of pairs
</Type>
<Description>   data points
</Description>
<Name>          $(w_i)_{i=1}^m$
</Name>
<Type>          Double: array ($m$)
</Type>
<Description>   (OPTIONAL) weights
</Description>
<Name>          $n$
</Name>
<Type>          Integer, $\ge 0$
</Type>
<Description>   order of fitting polynomial
</Description>
</Inputs>
<Outputs>
<Name>          $(A_{j})_{j=1}^n$
</Name>
<Type>          Double: array ($n$)
</Type>
<Description>   coefficients
</Description>
<Name>          Error
</Name>
<Type>          *
</Type>
<Description>   indicates presence/nature of error
</Description>
</Outputs>
<Aim>
                \[ (A_{j})_{j=1}^n \mbox{ minimises }
                \sum_{i=1}^m w_i^2 (y_{i} - P(x_{i}))^{2} \]
                where \[ P(x) = \sum_{j=1}^n A_{j} x^{j-1} \]
</Aim>
<Validation>    reference data-sets
</Validation>
</KF>
```

Figure 13: XML file for key function: Polynomial fit

The proper solution to mathematics on the WWW is being developed by the W3C (the World Wide Web consortium). They have defined a markup language for mathematics, MathML, that can be included in web pages. Unfortunately, MathML is not currently supported by the most common browsers although there are "plug-ins" available that allow browsers to display MathML. There is a good chance that new versions of Netscape or Internet Explorer will support MathML directly, in the next year.

METROS will use MathML for its mathematical specifications, when the language is widely supported. The existing key function specifications can be converted to MathML, as there are tools to convert LaTeX to MathML. Contributions to METROS would be able to include mathematics in LaTeX or other languages that can be translated to MathML, or the contributions could include MathML produced by the contributor. The use of MathML will provide a firm basis for the future use of mathematics in METROS.

A Key functions list

This is the list of key functions currently included in METROS, by category.

Evaluation functions — geometric: distance (2D), distance (3D), distance to line (2D), distance to line (3D), distance to plane, distance to circle (2D), distance to circle (3D), distance to sphere, distance to cylinder, distance to cone

Evaluation functions — series: evaluation polynomial (by powers), evaluation polynomial (Chebyshev), spline evaluation, Fourier series evaluation, sum of exponentials

Evaluation functions — simple statistics: sample mean, sample standard deviation, geometric mean, harmonic mean, Huber mean, Gaussian distribution, inverse Gaussian, inverse Student's t-distribution, inverse χ^2 distribution, inverse F-distribution

Evaluation functions — properties of materials: refractive index of air, partial water vapour pressure (humidity), partial water vapour pressure (dew point), density of air, density of water, ITS-90 temperature, mass correction

Solver routines: linear least-squares, linear ℓ_1, linear Chebyshev, non-linear least-squares, ODE solver, zero-finder

Statistics routines: standard variance (explicit), standard variance (implicit), expanded uncertainty (explicit, Gaussian), Monte-Carlo simulation (parametric), Monte-Carlo simulation (non-parametric)

Fitting routines: linear orthogonal distance regression, circle fit, polynomial fit, spline fit, Fourier series fit

B Acronyms

The following acronyms are used within this best practice guide.

AF	Application Function
BPG	Best Practice Guide
CS	Case Study
DTI	Department of Trade and Industry
EUROMET	European organisation of metrology institutes
EUROMETROS	European METROS
GKF	Generic Key Function
HTML	Hypertext Markup Language
IAF	Implementation of Application Function
IKF	Implementation of Key Function
INTMET	Interdisciplinary committee of EUROMET
ITS	International Temperature Scale
KF	Key Functions
MA	Metrology Area
MathML	Mathematical Markup Language
METROS	METROlogy Software environment
NAG	Numerical Algorithms Group
NMI	National Measurement Institute
NMS	(UK) National Measurement System
NPL	National Physical Laboratory
ODE	Ordinary Differential Equation
SS*f*M	Software Support for Metrology
W3C	World Wide Web consortium
WWW	Worldwide Web
XML	Extensible Markup Language