Enabling the information society

Presentation to BCS Coventry Branch

23rd March 2016

Fortran, alive and well at 59!

Peter Crouch PhD CEng FBCS CITP FIMF

Chairman BCS Fortran Specialist Group

pccrouch@bcs.org.uk, www.fortran.bcs.org

Presentation Outline

My Background

Early Years of FORTRAN

Fortran Pioneers – in Their Own Words

Fortran Standards Summary

BCS Fortran Specialist Group and its Activities

Compiler Support for Fortran Standards

Current Application Areas for Fortran

Further information & Acknowledgements

My Background

1968 – 1984	Industrial research chemist. Started programming in BASIC and Pascal in the late 1970s. Began to use Fortran in the early 1980s.
1985 – 2001	Software developer for Computer Aided Design and Manufacturing systems using Fortran and C.
2003 – 2005	Junior civil servant in the Department for Work and Pensions.
2008 – 2014	Part-time consultant to the Nickel Institute
1993	Joined the British Computer Society
1997 – 2002	Chairman of BCS Birmingham Branch
2002 – to date	Chairman of BCS Fortran Specialist Group
2015 – to date	Treasurer of BCS Wolverhampton Branch

Commodore PET from late 1970s on which I learned to program in BASIC

Apple II system from early 1980s on which I learned to use Pascal and Fortran

Early years of FORTRAN: 1954-1967

1954	**Development work starts in IBM led by John Backus**
1957	**IBM release a FORTRAN compiler for the IBM 704**
1958	**IBM release FORTRAN II, with subroutines and blank common**
1960	**Philco release ALTAC, a FORTRAN II look-alike**
1961	**IBM have eight different compilers (for the 709, 650, 1620 and 7090) and publish a guide to language variations between them**
1961	**Univac release FORTRAN I for the SS80, the first compiler called 'FORTRAN' for a non-IBM machine**

See www.fortran.bcs.org/2007/jubilee/implementations.php for a list of FORTRAN implementations from 1957 to 1967

An IBM 704 mainframe from the late 1950s (image courtesy of LLNL)

Pioneer Day June 9 1982 Houston, Texas

THE
HISTORY OF COMPUTING COMMITTEE
OF THE
AMERICAN FEDERATION
OF
INFORMATION PROCESSING SOCIETIES
HONORS

IBM

ON THE TWENTY-FIFTH ANNIVERSARY OF ITS DEVELOPMENT OF THE PROGRAMMING LANGUAGE

FORTRAN

25 YEAR=1957 • 1982 DO 25 CONTINUE — Pioneer Day — Houston, Texas

PRESENTED AT
PIONEER DAY
NATIONAL
COMPUTER
CONFERENCE
JUNE 9, 1982
HOUSTON, TEXAS

CHAIRMAN, PIONEER DAY — CHAIRMAN, NCC '82

CHAIRMAN, HOCC — PRESIDENT, AFIPS

The 25th Anniversary
of
FORTRAN
June 9, 1982
Pioneer Day
1982 National Computer Conference
1:45 p.m. The Early Days of FORTRAN
3:30 p.m. The Institutionalization of FORTRAN

FORTRAN
—"the infantile disorder"—
is hopelessly inadequate for whatever computer application you have in mind today . . . too clumsy, too risky and too expensive.
—Edsger Dijkstra

God is Real
(unless otherwise declared in an explicit type statement or in an implicit declaration).
—B. Graham

I don't know what the language of the year 2000 will look like but I know it will be called FORTRAN
—Tony Hoare

FORTRAN
is a collection of Warts, held together by bits of Syntax.
—Anon.

FORTRAN
is a language to avoid
—unless you want some answers
—Anon.

The one central attribute of FORTRAN is its name
—Martin Greenfield

In the good old days, physicists repeated each other's experiments, just to be sure.
Today they stick to FORTRAN so they can share each other's programs, bugs included.
—Edsger Dijkstra

Sayings from cards distributed at Pioneer Day.

Annals of the History of Computing, Volume 6, Number 1, January 1984 • 13

Fortran Pioneers, led by John Backus, reunited 25 years on - June 1982

Standing, from left: John Backus, Sheldon Best, Robert Nelson, Irving Ziller.
Seated, from left: Richard Goldberg, Lois Habt, Roy Nutt.

From left: Richard Goldberg, Robert Nelson, Lois Habt, Roy Nutt, Irving Ziller, Shelton Best, Harlan Herrick, John Backus, Peter Sheridan.

The creation of Fortran: in their own words

There are two Fortran films available for download from www.softwarepreservation.org/projects/FORTRAN/video

The first, from 1982, was commissioned by IBM mark the 25th anniversary of the release of the first Fortran compiler. It features John Backus and members of his development team describing how they created the first Fortran compiler between 1954 and 1957.

It is also available for viewing on YouTube.

The second, from 1958, is an IBM demonstration of using Fortran I or II to calculate compound interest. It is available in Quick Time and Windows Media (WMV) format. The video quality is rather poor.

The creation of Fortran: in their own words

John Backus (1924-2007), *centre*, 47 years on - February 2004 with Alex Stepanov, *left*, and Paul McJones

FORTRAN standards summary (1)

1966	**ANSI standard X3.9-66 (FORTRAN 66)** First programming language standard Developed between 1962 and 1966 Essentially a common subset of vendors' offerings **The US standard is reproduced as an ISO standard in 1972**
1978	**ANSI standard X3.9-78 (FORTRAN 77)** Introduced CHARACTER data type, IMPLICIT, PARAMETER, SAVE statements, IF - THEN - ELSE construct **Also published as ISO standard 1539:1980**
1990	**Development of Fortran 90** Originally scheduled for completion in 1982 Renamed Fortran 8X, then Fortran 88 and finally completed after rancorous discussions and attempts by some US vendors to derail the entire project

Fortran standards summary (2)

1991	**ISO/IEC standard 1539:1991 (Fortran 90)** Major revision - introduced free form source form, whole array operations, memory allocation at runtime, facilities for modular data and procedure definitions, parameterized intrinsic types, user-defined data types, pointers & many minor modernizations and removal of arbitrary restrictions

See "**The Fortran (not the foresight) saga: the light and the dark**" by Brian Meek and "**The Standards Hiatus**" by Miles Ellis and Lawrie Schonfelder, both linked from www.fortran.bcs.org/2007/jubileeprog.php, for more information on the development of Fortran 90

1997	**ISO/IEC standard 1539-1:1997 (Fortran 95)** Minor revision – introduced FORALL construct, PURE and ELEMENTAL procedures, initialization for pointers and for structures, some older, duplicated features designated as 'obsolescent'

Fortran standards summary (3)

2000	**ISO/IEC 1539-2:2000 - Varying length character strings** Revision of a previous version issued in 1994. Took account of improvements introduced in Fortran 95. Defined interface and semantics for module that provided facilities for manipulation of character strings of arbitrary and dynamically variable length.
2004	**ISO/IEC 1539-1:2004 (Fortran 2003)** Major revision - introduced object oriented programming support, standardised interoperability with C, I/O enhancements, including stream access and asynchronous transfers, access to command line arguments & environmental variables, support for IEEE arithmetic & exception handling and for ‘international usage’
2010	**ISO/IEC standard 1539-1:2010 (Fortran 2008)** Major revision - introduced coarrays as an extension for parallel processing, execution of command shell commands, I/O enhancements including getting unique unit numbers, new edit descriptors, BLOCK construct with declarations

Fortran standards summary (4)

Two Technical Specifications, subsidiary standards, issued

2012

ISO/IEC TS 29113:2012 - Further Interoperability of Fortran with C

Provided for interoperability of interfaces with Fortran dummy arguments that are assumed-shape arrays, have assumed character length, or have the ALLOCATABLE, POINTER, or OPTIONAL attributes. New Fortran concepts of assumed type and assumed rank introduced.

2015

ISO/IEC TS18508:2015 - Additional Parallel Features in Fortran

Introduced concept of TEAMS of images, handling of FAILED images and posting of EVENTs by one image to notify another image. New generic intrinsic procedures specified and extensions specified for existing intrinsic procedures.

2018

ISO/IEC standard 1539-1:2018 (Fortran 2015)

Minor revision - incorporating the technical corrigenda to Fortran 2008, the two technical specifications (on further interoperability with C and on additional parallel facilities), editorial improvements and removal of existing deficiencies and irregularities.

The BCS Fortran Specialist Group is established: 1970

FSG Minutes of 6 January 1970:

The objectives of the group were formally agreed to be:

(a) to form a focus in the United Kingdom for work concerned with establishing and maintaining FORTRAN standards.

(b) to work in association with national and international standardisation bodies.

FSG Minutes of 5 April 1976:

4. Revision of objectives

Following further discussion, the wording of the proposed revised objectives now becomes "To undertake activities associated with any aspects of Fortran".

It is intended to present this for approval at the next Specialist Groups meeting.

Fortran SG Activities 1970-1992

- Fortran SG hold typically four to six meetings per year, mostly discussing working party progress, applications, software tools, programming techniques and, from late 1971, Fortran standards developments
- From late 1970s FSG members become involved in development of Fortran standards and play a significant part in development of Fortran 90
- Presentations are made at conferences and workshops, e.g. Datafair 73, 75 and 77 and a Fortran Forum in London in 1978 with six US members of X3J3 as speakers
- Fortran Forums are held in London (4) and Edinburgh (2), sometimes with visiting US speakers, other meetings are held outside London
- See www.fortran.bcs.org/archive.php for more details of Fortran SG activities from 1970 to 1992

Fortran SG Activities 1993-2007

- In mid-1990s attendances wilt with the advent of the internet. It is decided to hold only annual meetings plus special events
- The Fortran SG is revivified. Events are held in 2002 and 2005 to discuss UK requirements for inclusion in future Fortran standards
- From 2002 successful applications are made to BCS to support between one and three FSG members to attend ISO WG5 meetings to help put the UK case on Fortran standards
- In 2007 a very successful full-day meeting is held with the Computer Conservation Society to mark the 50th anniversary of the release of the first Fortran compiler
- FSG members organized the 2007 ISO WG5 meeting in BCS London offices and held a reception for WG5 delegates
- See www.fortran.bcs.org /2007/jubileevents.php for more details of Fortran activities in 2007

'Fifty Years of Fortran' meeting
January 2007

FSG Reception for WG5 delegates and partners on evening of 6 August 2007

held in The Weston Room at King's College Maughan Library

Fortran SG Activities 2010-2015

- In 2010 a successful meeting was held with the Institute of Physics Computational Physics Group to mark the 40th anniversary of the Fortran SG
- From 2011 the FSG AGM has been followed by an afternoon of Fortran-related talks including updates on the progress on Fortran standardisation, organised with the IoP CPG
- FSG members organized the 2015 ISO WG5 meeting in BCS London offices and a dinner for delegates and partners was held at the Salieri Restaurant in the Strand
- See www.fortran.bcs.org/pastevents.php for more details of Fortran SG activities since 1993

WG5 delegates and partners outside BCS London office on 5 August 2015

before going to dinner at the Salieri Restaurant in the Strand

Compiler support for Fortran 2003 & 2008 Standards

Table first published in April 2007, revised twice a year. Compiled by Ian Chivers and Jane Sleightholme, www.fortranplus.co.uk

Section on Fortran 2008 features added in August 2009

Section on TS 29113 features added in December 2013

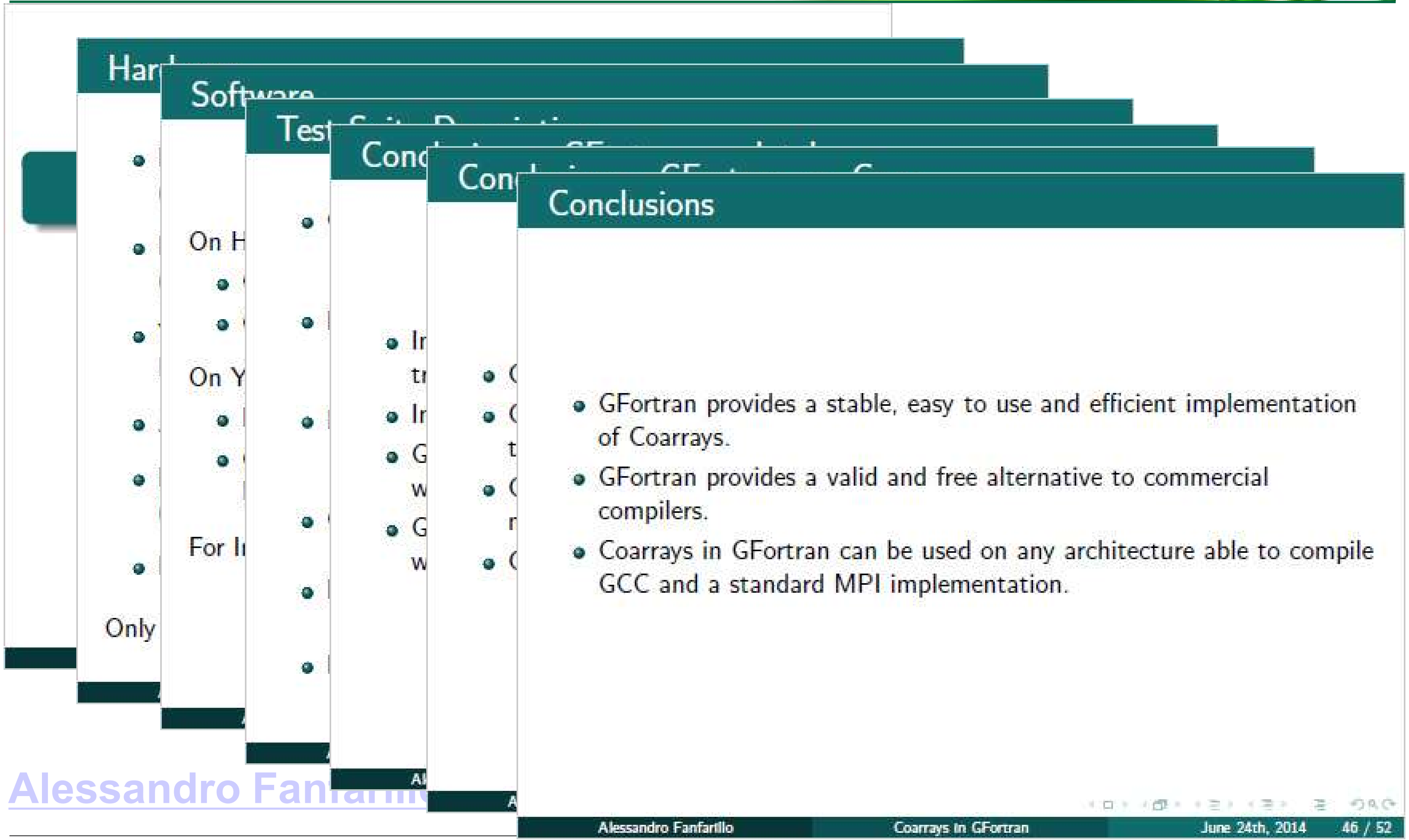
Information on 11 compilers is currently available

Latest version published in ACM Fortran Forum magazine, Revision 17 – November 2015

Previous version available on Fortranplus website at www.fortranplus.co.uk/resources/fortran_2003_2008_compiler_support.pdf , Revision 16 – June 2015

Fortran Compiler Comparisons are available from the Fortran UK website at www.fortran.uk/fortran-compiler-comparisons-2015/

Report presented to WG5 meeting in Las Vegas, June 2014



Some current application areas for Fortran

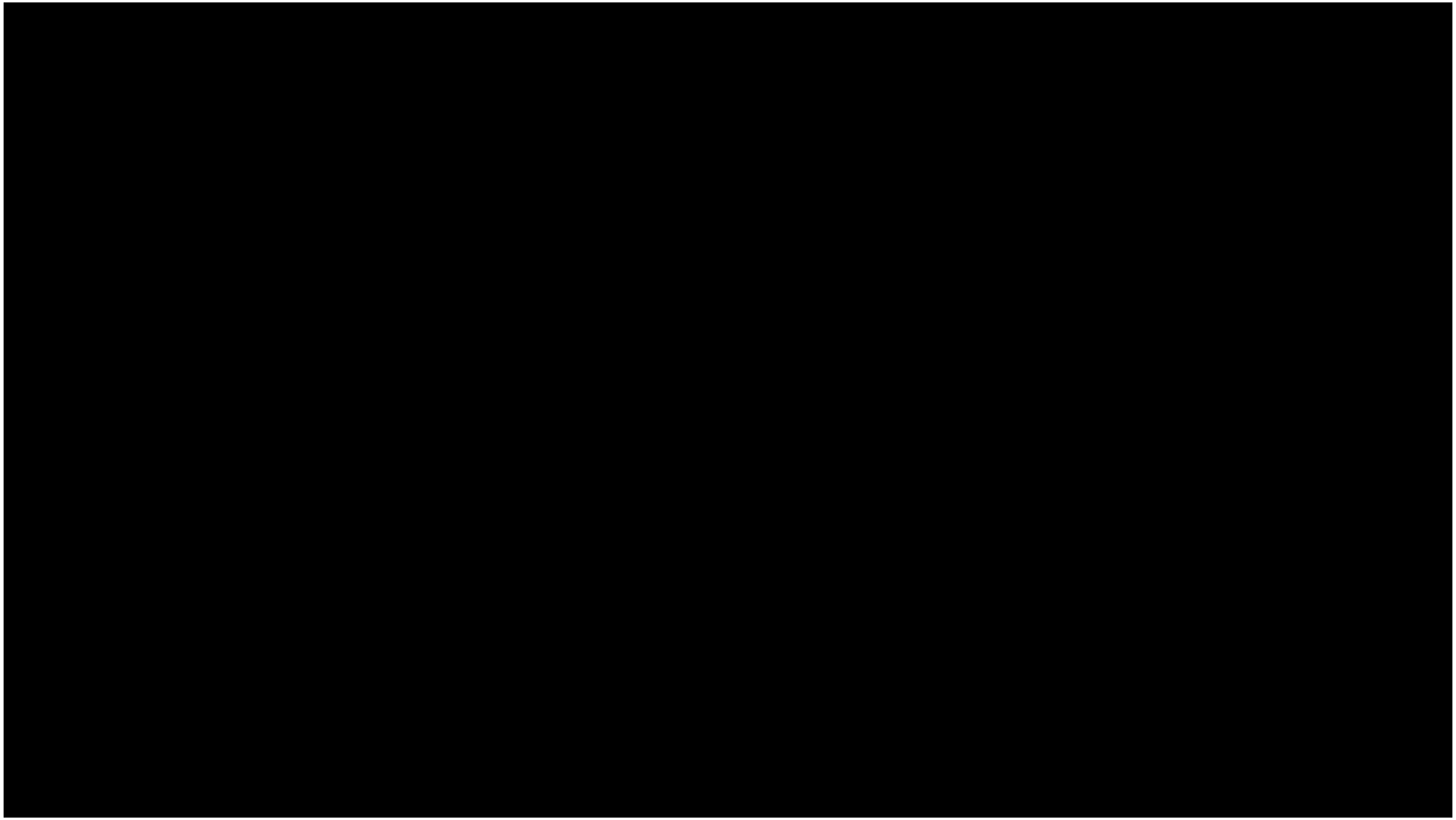
- Weather forecasting and climate prediction
- Analysis of seismic data for oil and gas exploration
- Nuclear test ban verification (forensic seismology)
- Modelling of nuclear weapons
- Financial analysis
- Vehicle crash simulation
- Computational fluid dynamics (CFD)
- Mathematical modelling of materials and processes
- Computerised aircraft performance monitoring

Supercomputing in the USA, 2010

DreamWorks Presents the Power of Supercomputing

In partnership with the Council on Competitiveness, Dreamworks produced a short animated film on the importance of high performance computing to the U.S. economy. Most of the applications described are written in Fortran.

Supercomputing video © 2011



Supercomputing at the UK Met Office

Met Office supercomputing

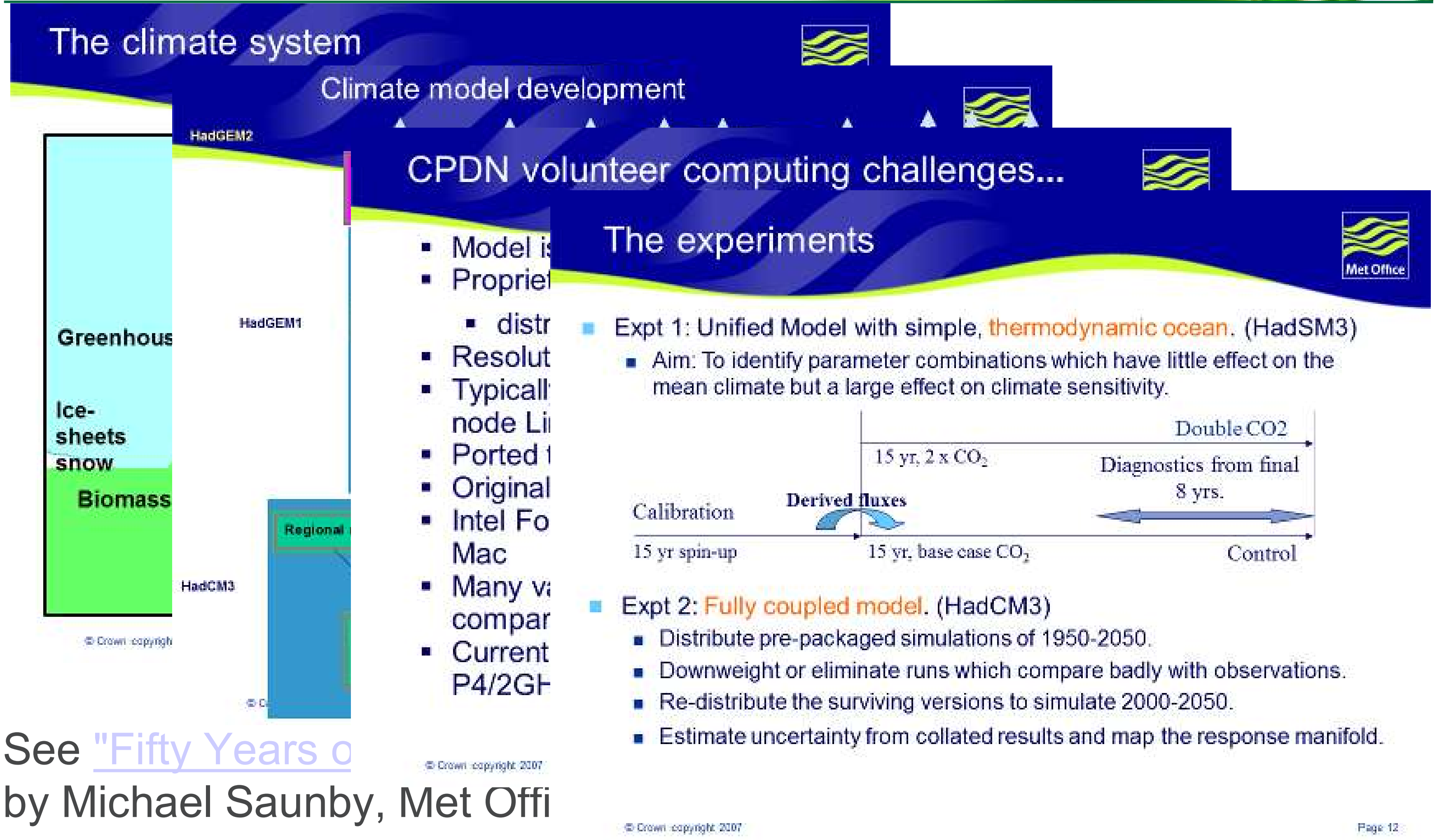
	SX-6	SX-8
CPU/node	8	8
Memory	32GByte	64GByte
Peak/node	64GFlops	128GFlops

3 clusters –

- 15 x SX-6
- 19 x SX-6
- 21 x SX-8

Weather forecast for Monday March 7 2016 produced by the UK Met Office

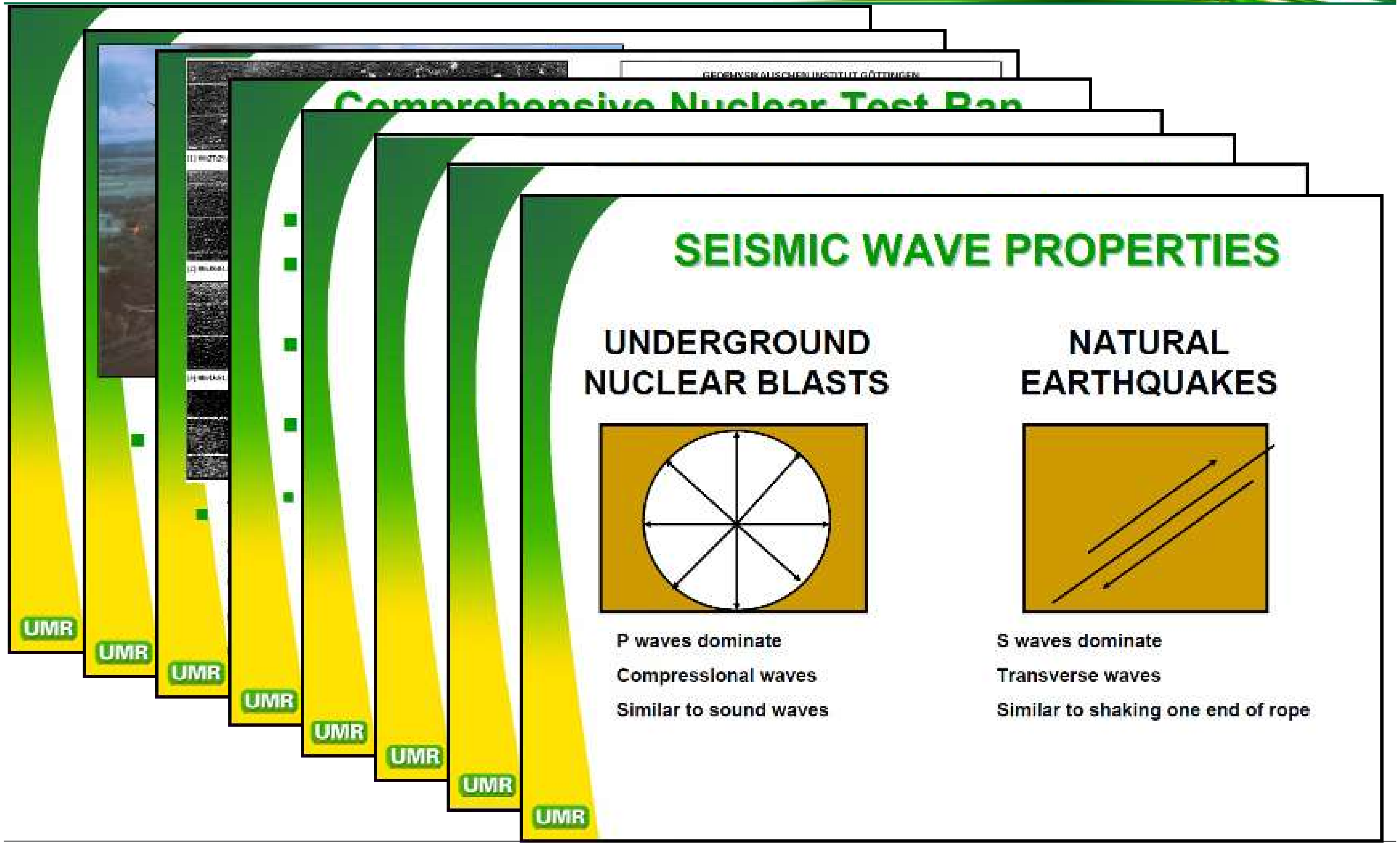
The Met Office climate model HadSM3 and climateprediction.net



See "Fifty Years o

by Michael Saunby, Met Offi

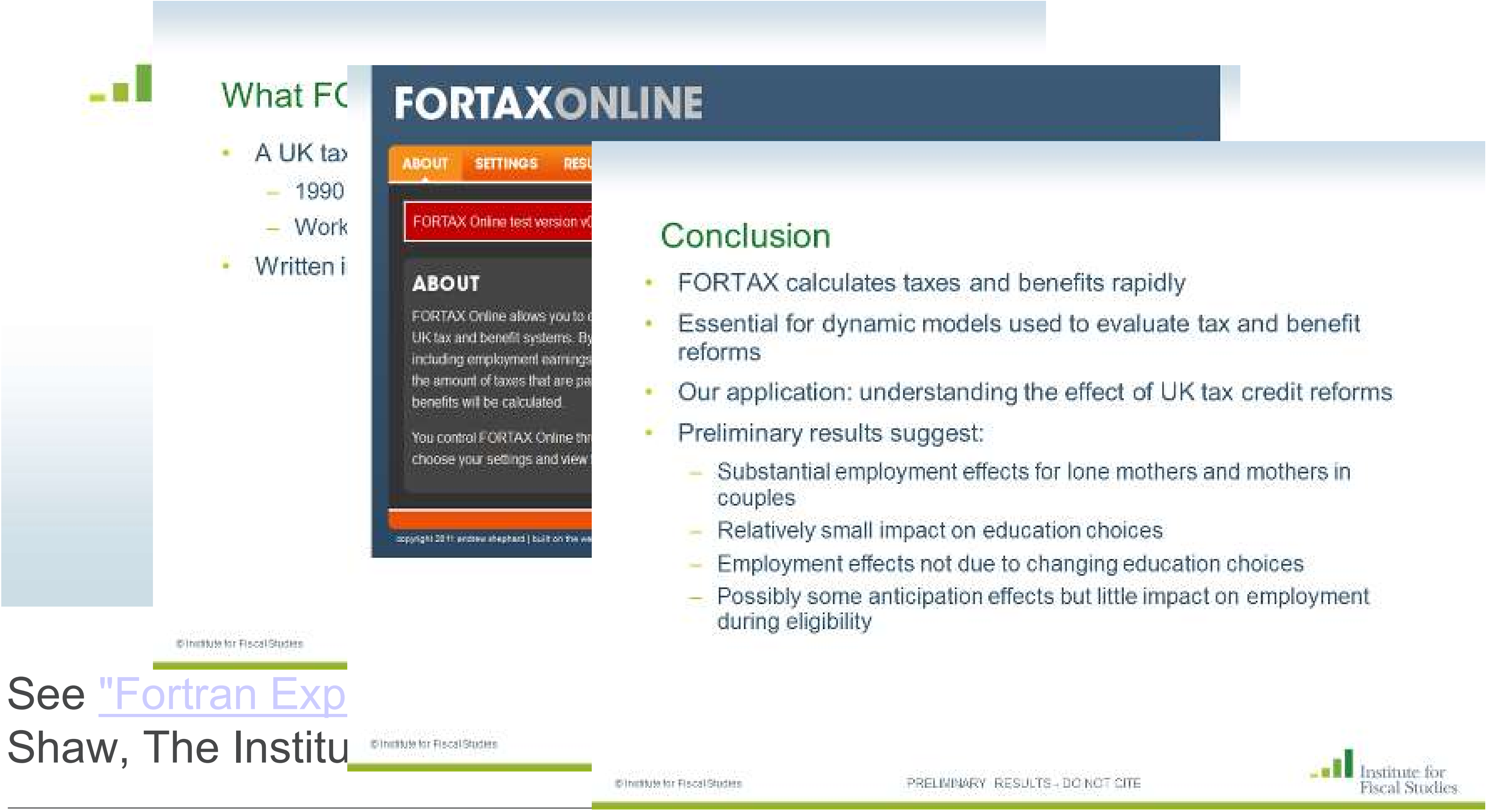
Forensic Seismology for Nuclear Test Ban Verification



More information on Forensic Seismology and Nuclear Test Ban Verification

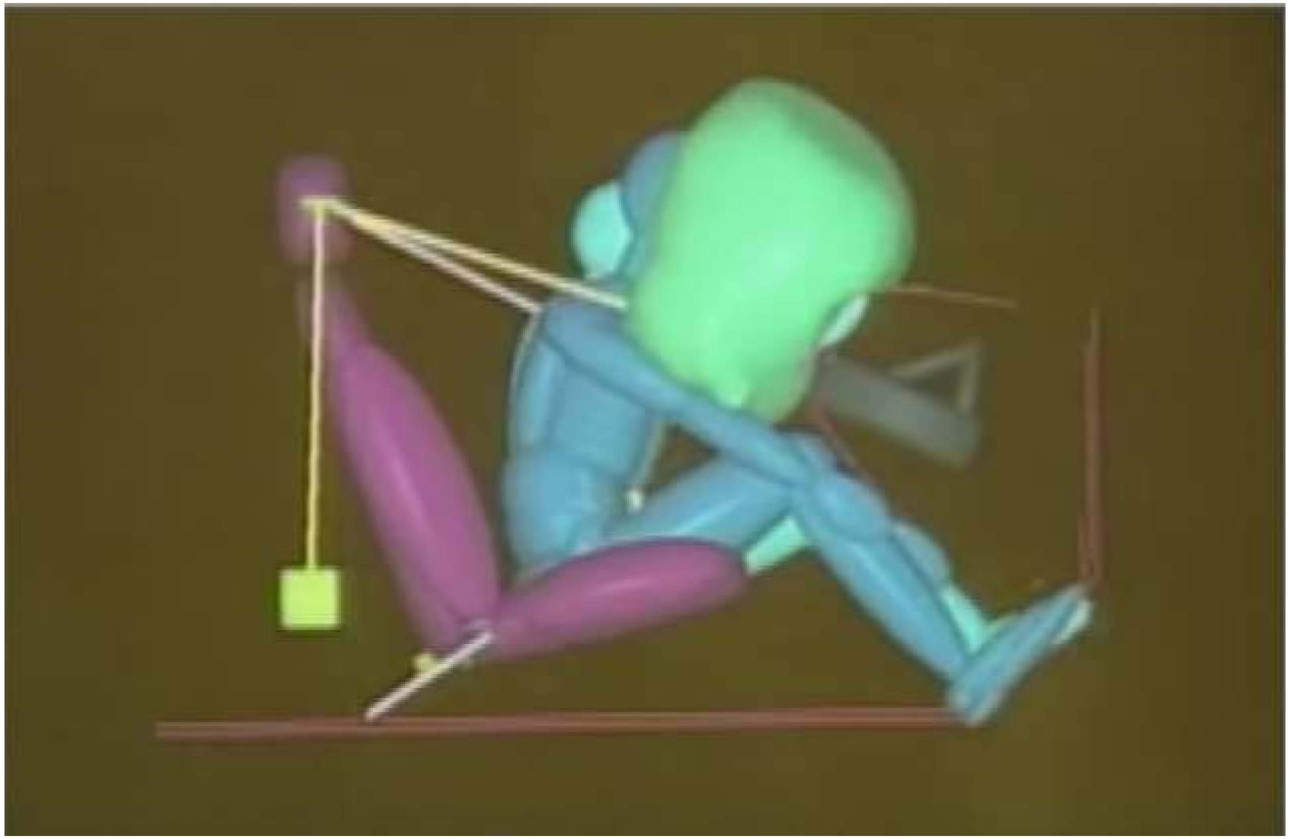
September 1998	Forensic Seismology Supports the Comprehensive Test Ban Treaty **(HTML)**
Early 2000s	Some Practical Applications of Forensic Seismology **(PDF)**
January 2007	Contributions by T.L. van Raalte and John Young to “Fifty Years of Fortran” meeting **(PDF & HTML)**
January 2009	Forensic Seismology and the Comprehensive Nuclear-Test-Ban Treaty **(PDF)**
November 2011	Supercomputers offer tools for nuclear testing — and solving nuclear mysteries **(HTML)**

FORTAX: a tax and benefit micro-simulation library written in Fortran

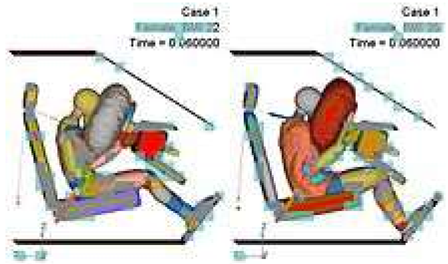


See "Fortran Exp
Shaw, The Institu

Vehicle crash simulation



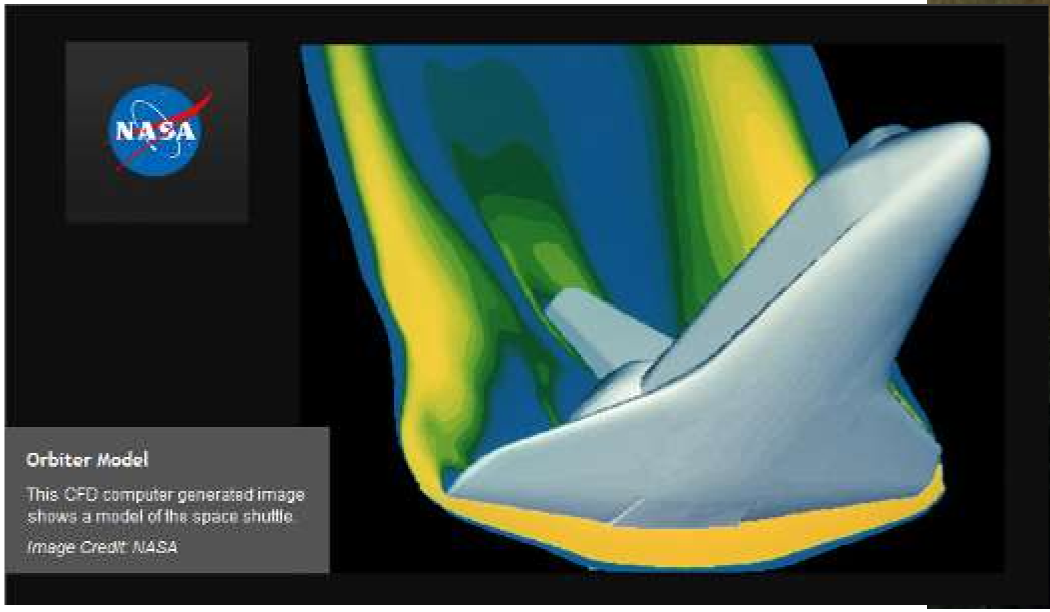
A crash simulation with a slender (left) and obese (right) female passenger



Harvard Lomax, 1922 – 1999

The father of
"The Numerical Windtunnel"

NASA Ames Research Center,
Moffett Field, California,
1944 – 1994

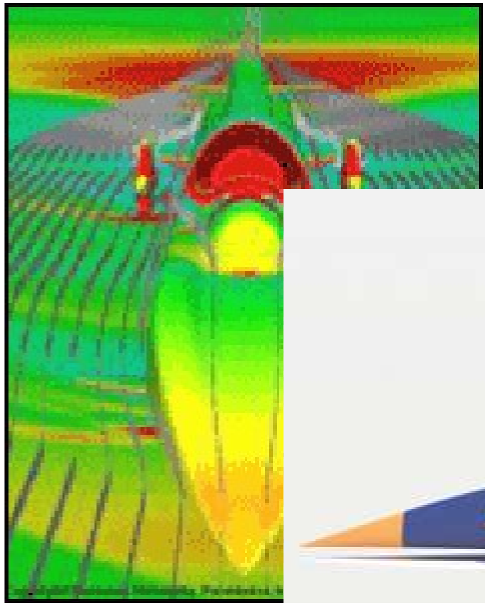


Fortran CFD program used in design of 1000 mph car - September 2010 issue of ITNOW

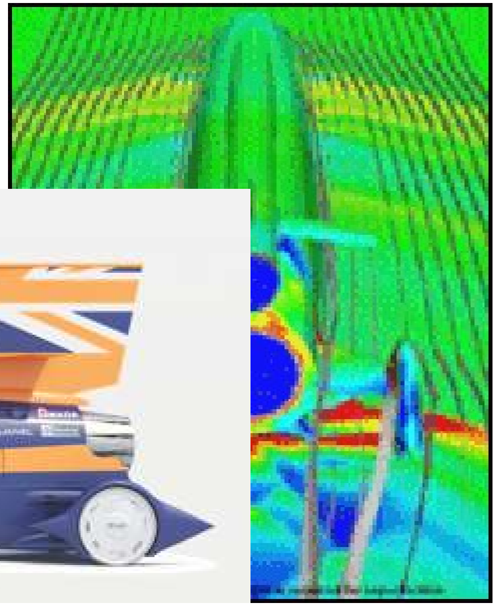
[The BLOODHOUND Project]

BLOODHOUND SSC, the 1000 mph car

The Super Sonic Car has been aerodynamically designed by Swansea University's School of Engineering experts who have pioneered Computational Fluid Dynamics (CFD)



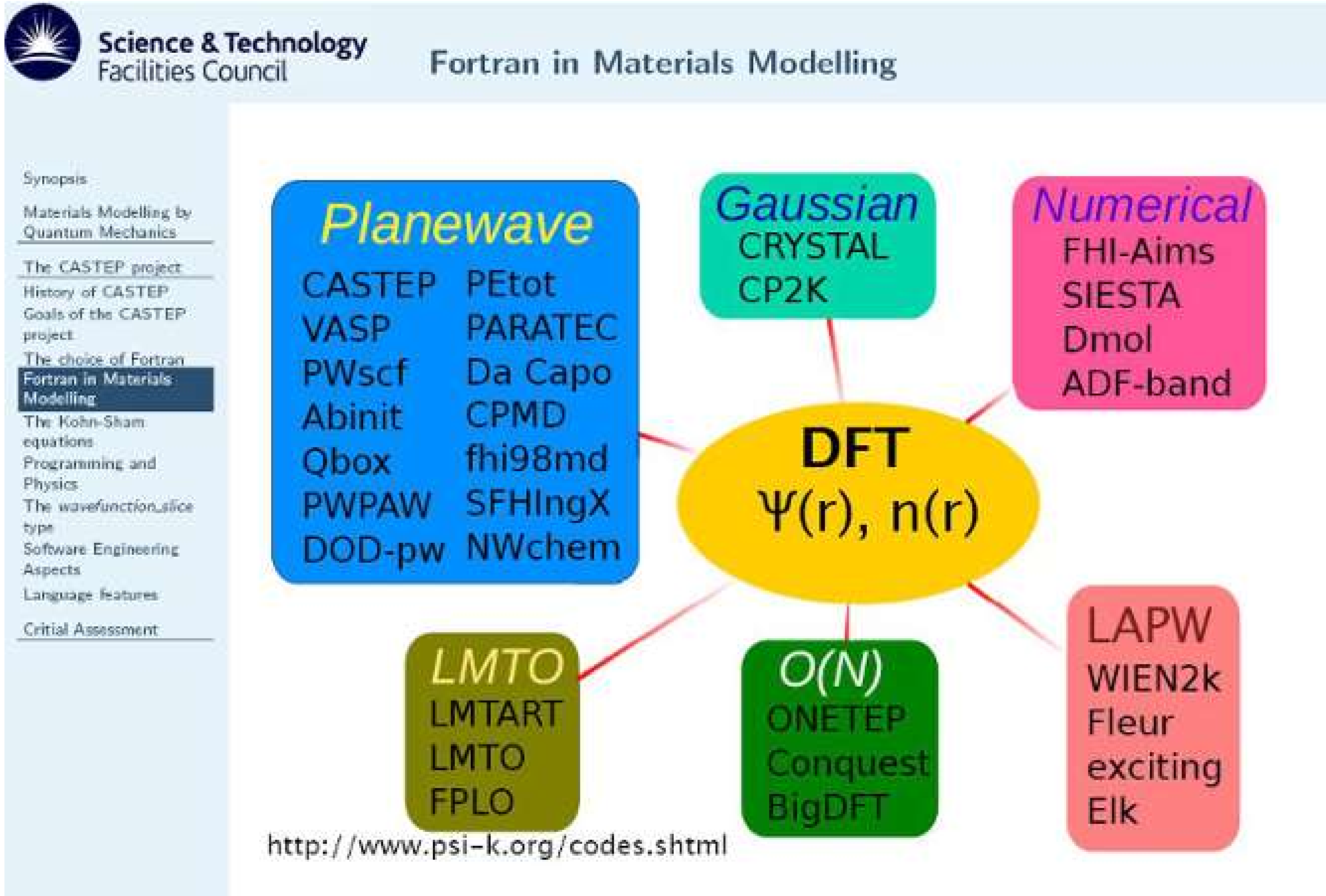
Front view



Rear view

The BLOODHOUND team plans to go to Hakskeen Pan in Northern Cape Province, South Africa, in October 2016, following successful runway testing in the UK in Spring 2016.

The CASTEP project:
Materials Modelling by Quantum Mechanics



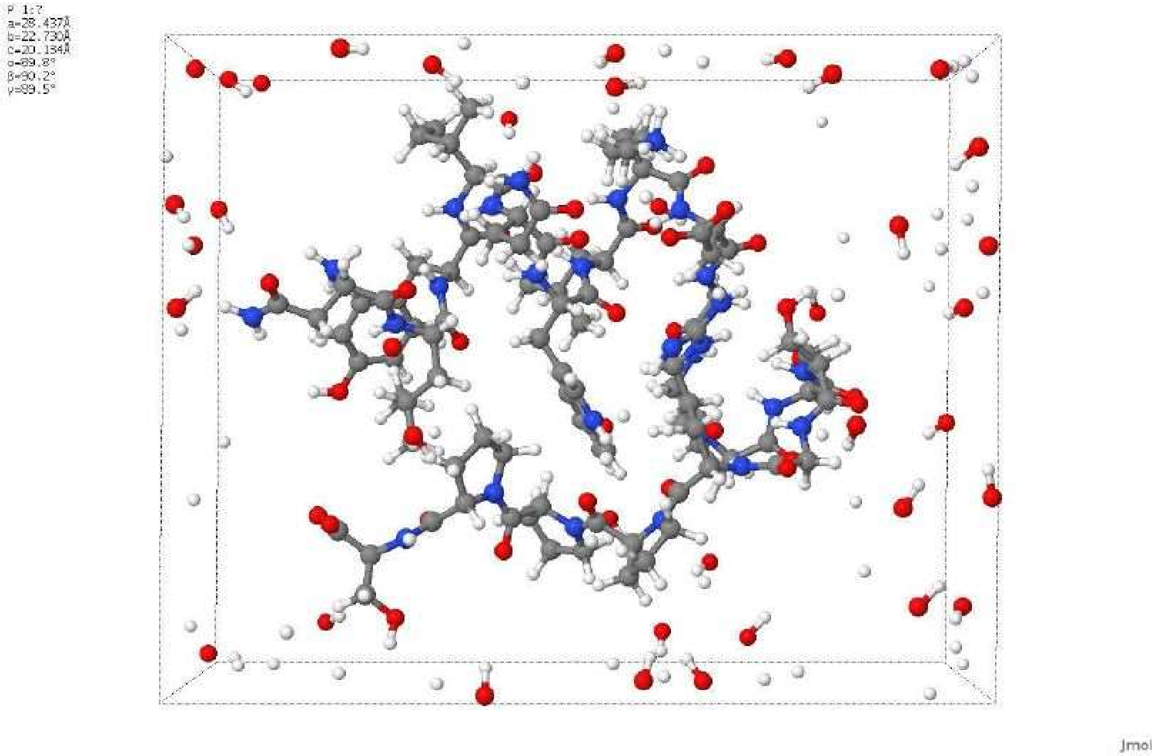
Keith Refson, Rutherford Appleton Laboratory, full presentation can be downloaded from www.fortran.bcs.org/2010/KR_BCS_2010_web.pdf

CASTOR HPC Capability

Science & Technology Facilities Council

HPC Capability: Peptide in water 1280 atoms

- Synopsis
- Materials Modelling by Quantum Mechanics
- The CASTEP project
- Critial Assessment
- CASTEP Features
- Citation report
- Azobenzene as a molecular switch
- **HPC Capability: Peptide in water 1280 atoms**
- HECToR Performance
- Retrospective assessment of design architecture
- Co-evolution of Fortran and CASTEP
- Unsatisfactory aspects
- Summary



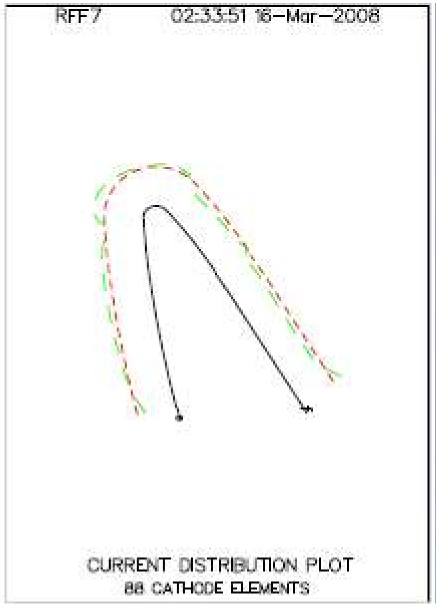
HyperSizer and Virgin Atlantic GlobalFlyer

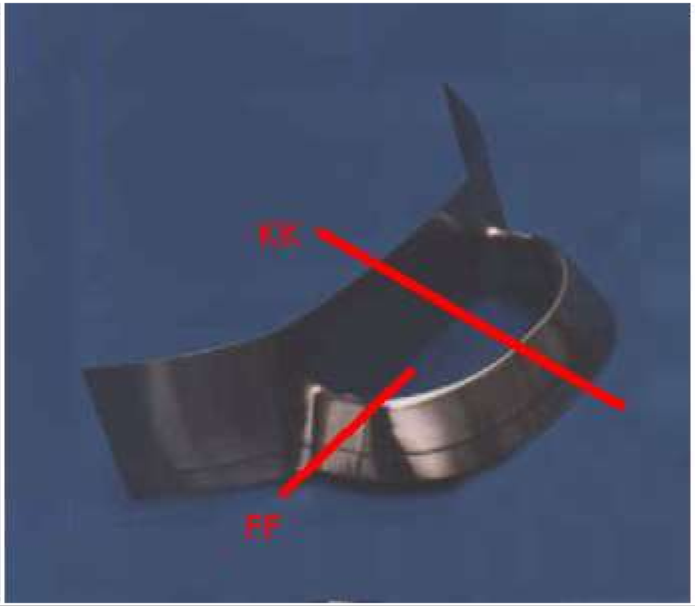
HyperSizer, 400,00 lines of Fortran and Visual Basic FEA code used to optimise composite materials built into GlobalFlyer which flew non-stop around the world in 67 hours in February – March 2005 piloted by Steve Fossett.

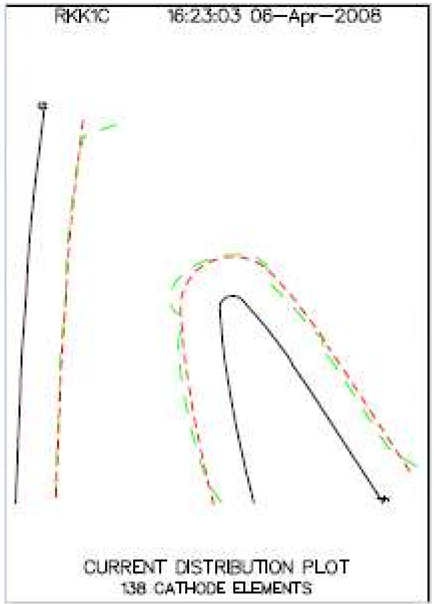
Boundary Element Package for Current Distribution Modelling in Electroforming

2D Boundary element program written in FORTRAN 77 for current distribution modelling in electrodeposition was developed in the early 1980s in the Department of Electrical Engineering at the University of Sheffield.
Program used in late 1980s and 1990s in the development of the production process for the manufacture of aerospace components in electroformed nickel.

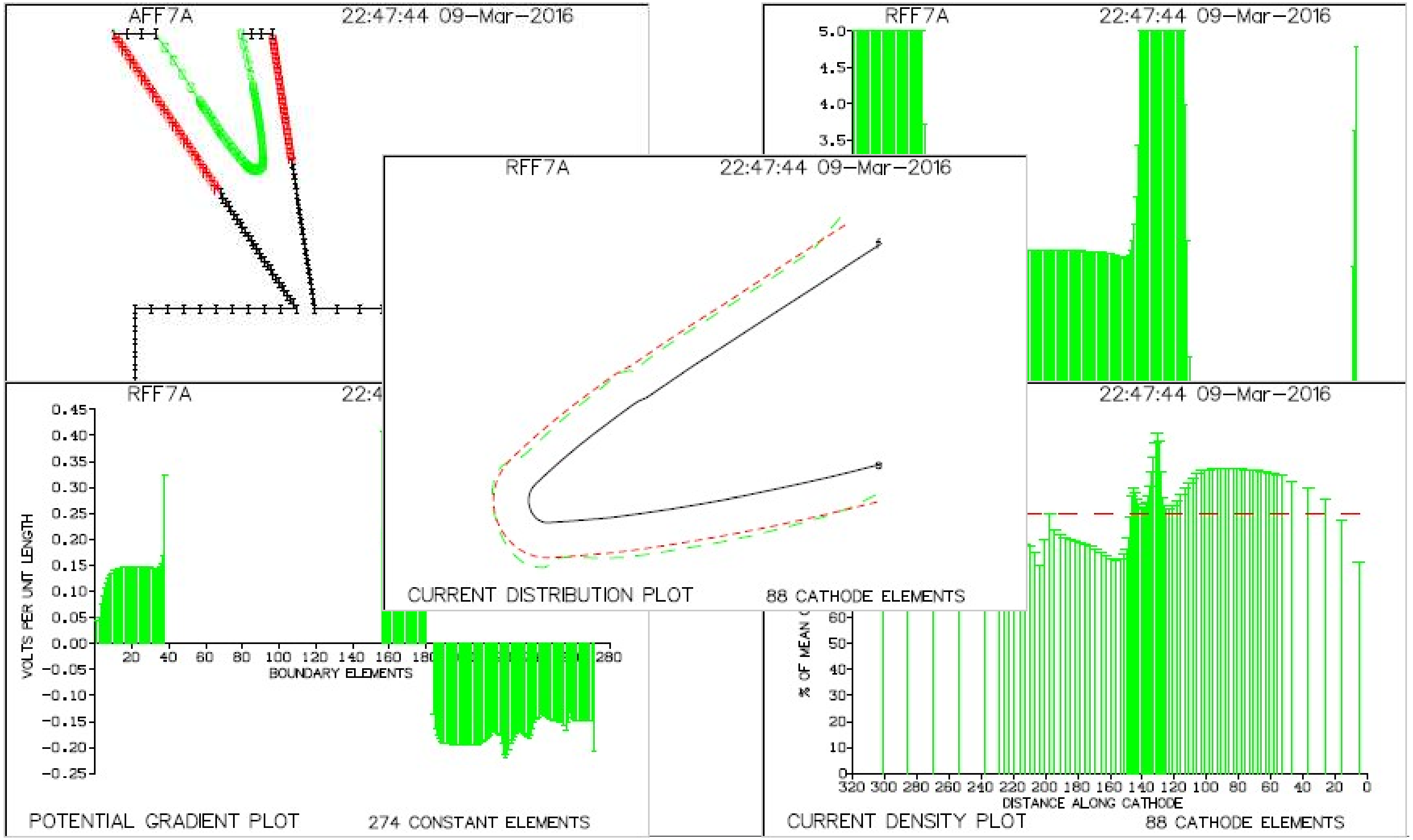
Modelling results for C130J R-R engine oil cooler lip







Modelling results for section FF of C130J R-R engine oil cooler lip



Nickel erosion shields manufactured by electroforming process

Components for:

Lockheed Martin C130J

Westland Lynx

Augusta Westland AW101, Merlin

Close-up of Rolls-Royce AE 2100 D3 turboprops with Dowty R391 composite scimitar propellers on a RAF Hercules C4 (C-130J-30)

G-LYNX, holder of world helicopter speed record, 400.87 km/h (249.09 mph)

Achieved on August 11 1986 using composite rotor blades protected with nickel and titanium erosion shields.
Record still had not been beaten by 2014.

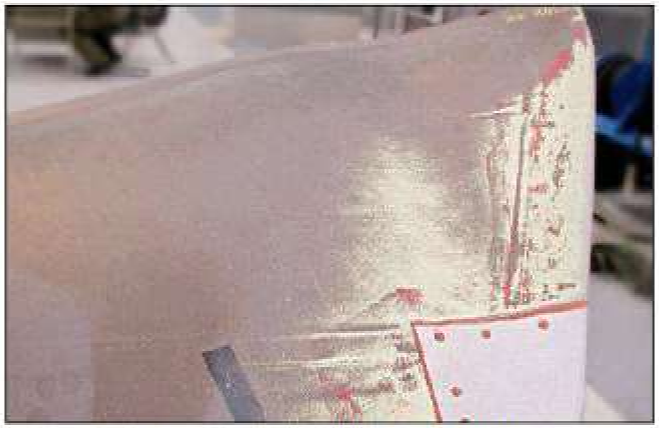
Headline from
FLIGHT INTERNATIONAL
27 December 1986

Fastest blades in the world

Last August a modified Westland Lynx broke the world helicopter speed record by a handsome margin. A speed of just over 216kt was reached, which meant that the advancing tips of its composite blades were near-sonic at Mach 0·97. **Harry Hopkins** reports.

Lynx and AW101 BERP rotor blades, RAF Merlin and BERP IV demonstrator

AW101 Blade Tip



Sand Erosion on Westland Lynx Blade Tip

Tip Strike on AW101 Blade (Crown Copyright)

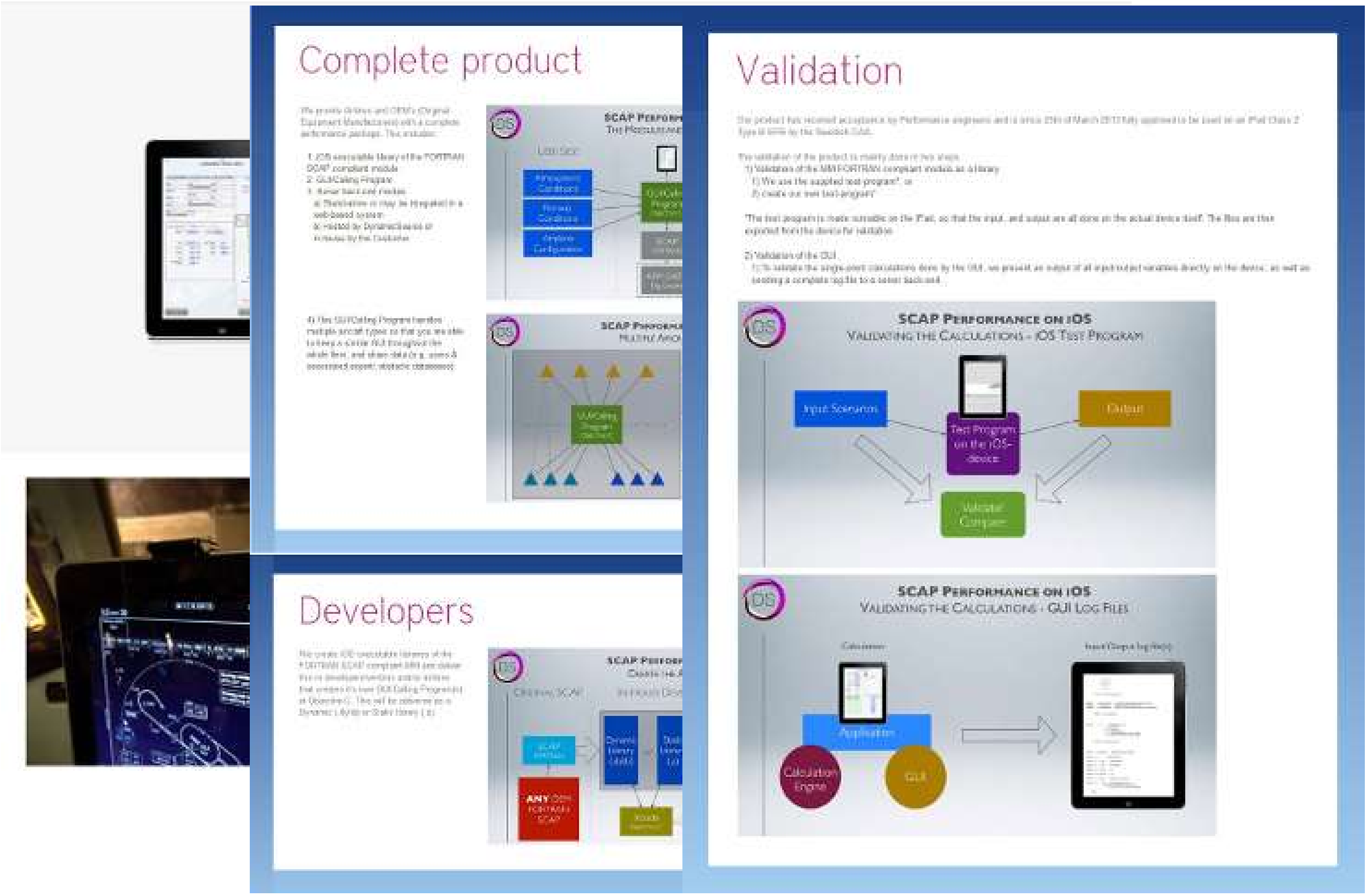
AW101, Royal Air Force Merlin Mk3

Exploitation is Next Step for BERP IV

BERP IV THE DESIGN, DEVELOPMENT AND TESTING OF AN ADVANCED ROTOR BLADE

RAF Hercules, Dutch Lynx and Royal Navy Merlin in flight

Fortran SCAP for iOS by DynamicSource AB (Standard Computerised Airplane Performance)



If you want to know more

Modern open source and free Fortran compilers are available from a number of sources as are online tutorials

Links to the above and more are available from the Resources page of the Fortran SG website at www.fortran.bcs.org/resources.php

"*The Seven Ages of Fortran*", a history of Fortran development with examples of modern Fortran concepts by Michael Metcalf , see http://journal.info.unlp.edu.ar/journal/journal30/papers/JCST-Apr11-1.pdf

"*Modern Fortran Explained*", Metcalf, Reid & Cohen, OUP, April 2011 See http://ukcatalogue.oup.com/product/9780199601424.do

"*Introduction to programming with Fortran: with coverage of Fortran 90, 95, 2003, 2008 and 77*", Ian Chivers & Jane Sleightholme, Springer-Verlag, 2015, see www.springer.com/gb/book/9783319177007

Further Information

FSG website
www.fortran.bcs.org/

WG5 document archive
www.nag.co.uk/sc22wg5/

J3 document archive
www.j3-fortran.org/

Fortran and Fortran II history, including 1958 & 1982 IBM films www.softwarepreservation.org/projects/FORTRAN/

Acknowledgements

Thanks are due to the many contributors to meetings of the Fortran Specialist Group between 2007 and 2014, including the 'Fifty Years of Fortran' meeting in January 2007 and the joint BCS/IoP meeting in June 2010 marking the 40th anniversary of the Group as many of the slides in this presentation are taken from their presentations.

Thanks also to Paul McJones of the Computer History Museum, Mountain View, CA, for providing me with DVD versions of two IBM films and the 2004 photo of John Backus.

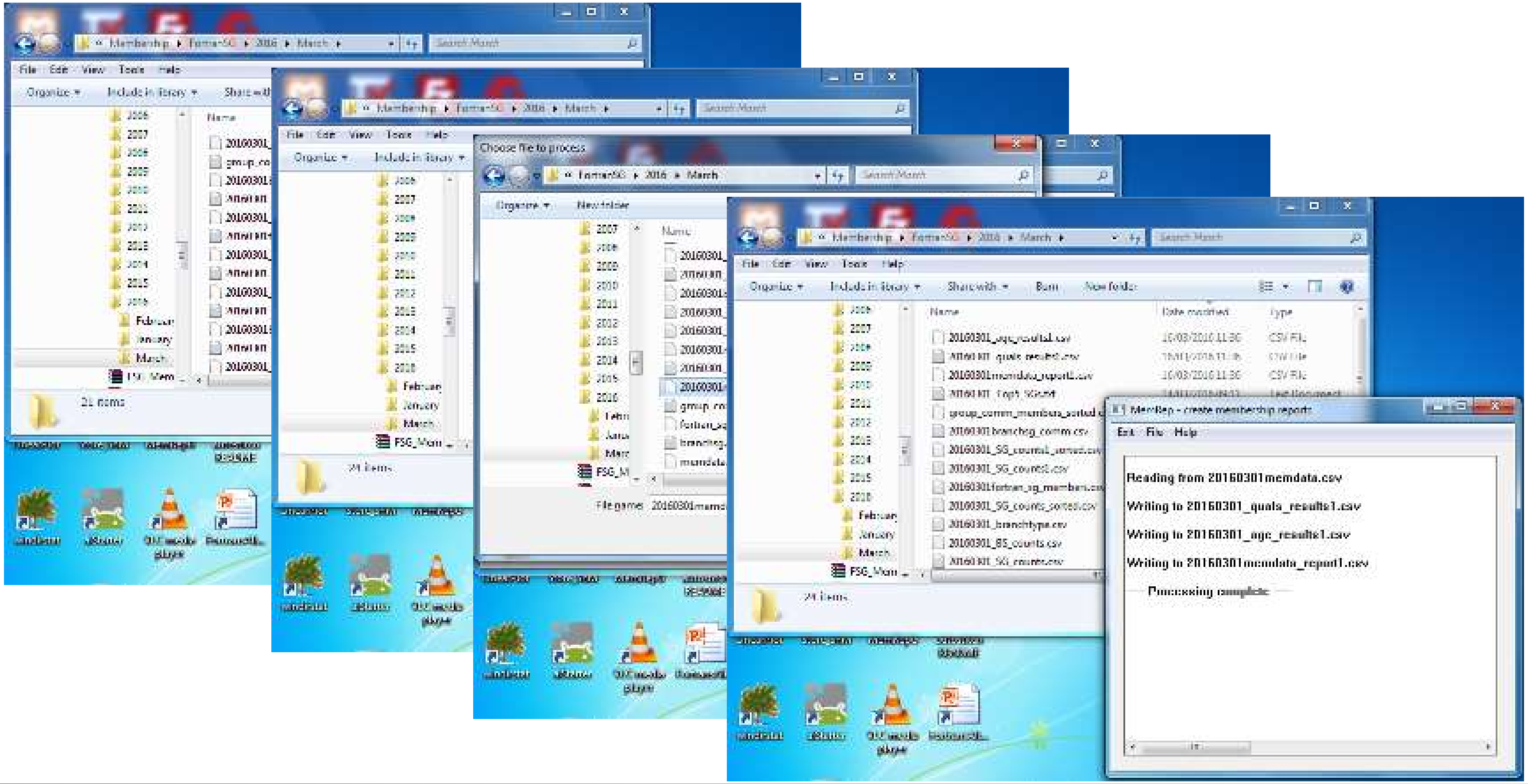
I also thank all my colleagues in the Fortran Specialist Group for their assistance and encouragement during my time as Chairman.

Finally thanks are due to BCS - the Chartered Institute for IT for their continued support of the UK's involvement in the ISO Fortran standardisation process through the Fortran Specialist Group and for hosting the 2007 and 2015 ISO Working Group 5 meetings in the BCS London Office.

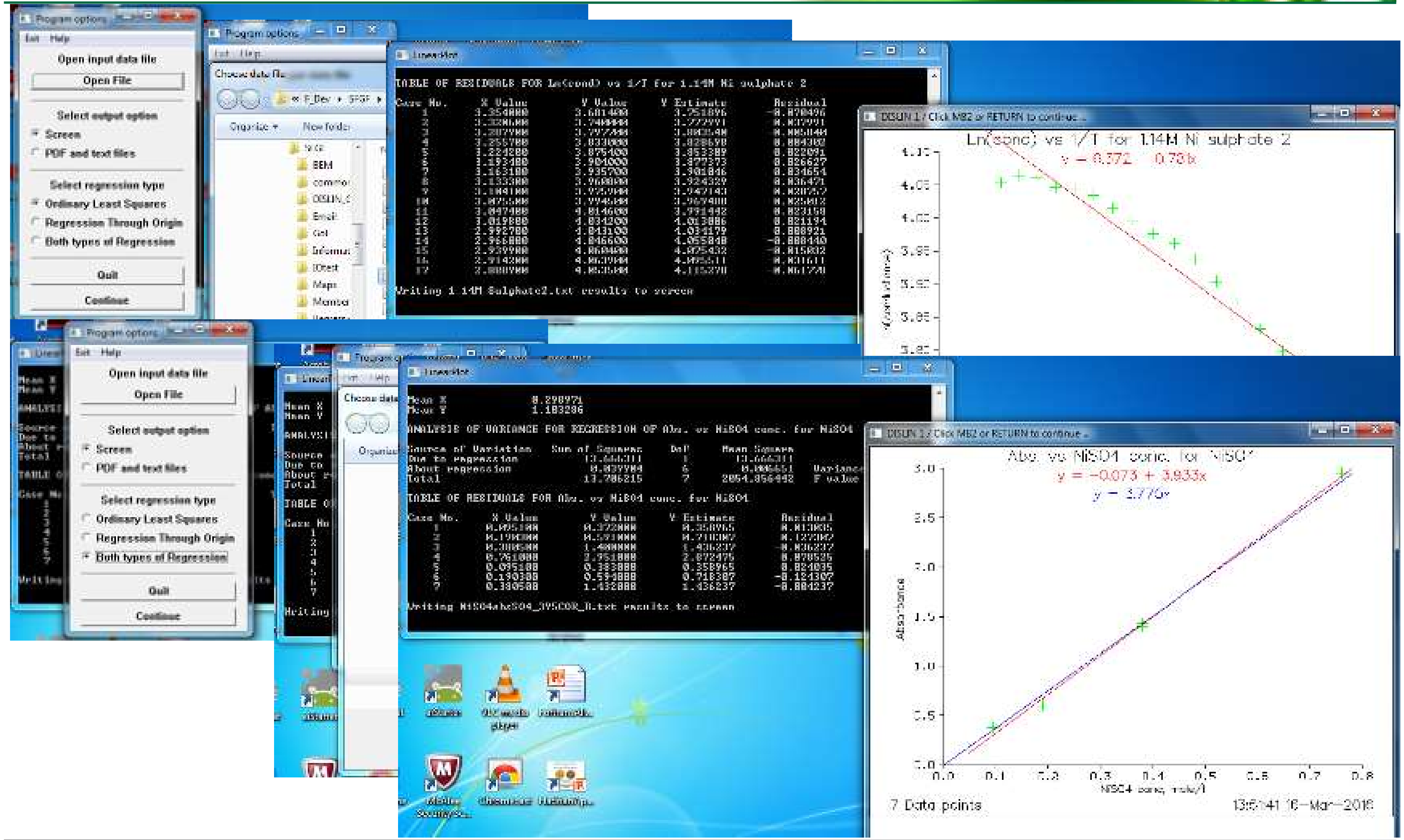
Any Questions?

Using the DISLIN library to create a GUI for Fortran programs (1)

Built using the Approximatrix Simply Fortran IDE with the GNU Fortran (GFortran) compiler and the DISLIN and MinGW graphics libraries



Using the DISLIN library to create a GUI for Fortran programs (2)



Modern Fortran example code (1)

```
program linear
! Program to calculate simple linear regression
! Reads input from data file, writes output to screen
  use file_read
  use reg_calc
  implicit none
  integer           :: nval
  character(len=64) :: file_name
  ! Get the name of the input file from the command line
  if (command_argument_count() >= 1) then
    call get_command_argument(1, file_name)
    ! Open input file and read data into allocated arrays
    call read_file(trim(file_name), nval)
    ! Calculate regression and display results
    if (nval > 0) then
      call calc_reg(nval)
    end if
  end if
end program linear
```

Modern Fortran example code (2)

```
module file_read
  public :: read_file
contains
  subroutine read_file(file_name, nval)
  ! Open data file and read in number of observations and  x and y data
    use data_store
    implicit none
    character(len=*), intent(in) :: file_name
    integer, intent(out)          :: nval
    integer, parameter            :: in_unit = 10
    nval = 0
    open(unit=in_unit, status="old", action="read", file=file_name, &
    position="rewind")
    read(unit=in_unit, fmt=*) nval
    allocate(xvals(nval), yvals(nval)) ! Allocate space for x and y data
    read(unit=in_unit, fmt=*) xvals, yvals
    close(unit=in_unit)
  end subroutine read_file
end module file_read
```

Modern Fortran example code (3)

```
module kinds
! Declaration of real data type with 15 digits of precision and
! range of 10^-307 to 10^+307
  implicit none
  integer, parameter, public :: double = selected_real_kind(15, 307)
end module kinds


module data_store
! Declarations of arrays used to store data for linear regression
  use kinds, only : double
  implicit none
  real(kind=double), dimension(:), allocatable, public :: xvals, yvals
end module data_store
```

Modern Fortran example code (4a)

```
module reg_calc
  public :: calc_reg
contains
  subroutine calc_reg(nval)
  ! Calculate linear regression for yvals upon xyals
  ! i.e. y = A + Bx where A is the intercept on the Y axis and B is the
  ! slope of the best fit straight line
    use data_store
    use kinds, only : double
    implicit none
    integer, intent(in) :: nval
    integer             :: i, dastat
    real(kind=double)   :: sumxy, sumxsq, sumysq, ssdureg, ssabreg, intercept, &
                           slope, xbar, ybar, percent, meansq, fvalue, yest
    character(len=11)   :: flabel
    ! First calculate means for x and y
    xbar = sum(xvals) / nval
    ybar = sum(yvals) / nval
    ! Replace original data with its deviation from means
    xvals = xvals - xbar
    yvals = yvals - ybar
```

Modern Fortran example code (4b)

```
! module reg_calc continued
    ! Calculate the corrected sums of squares and products
    sumxy = 0.0_double
    sumxsq = 0.0_double
    sumysq = 0.0_double
    do i = 1, nval
      sumxsq = sumxsq + xvals(i) * xvals(i)
      sumysq = sumysq + yvals(i) * yvals(i)
      sumxy = sumxy + xvals(i) * yvals(i)
    end do
    ! Now calculate regression parameters
    slope = sumxy / sumxsq
    intercept = ybar - slope * xbar
    ssdureg = (sumxy * sumxy) / sumxsq
    ssabreg = sumysq - ssdureg
    percent = (100.0_double * ssdureg) / sumysq
    meansq = ssabreg / (nval - 2)
```

Modern Fortran example code (4c)

```
! module reg_calc continued
    ! Variance ratio (F value) always calculated with larger estimate in the numerator
    if (ssdureg > meansq) then
      fvalue = ssdureg / meansq
      flabel = "    F value"
    else
      fvalue = meansq / ssdureg
      flabel = "    F' value"
    end if
    print "(/,a,f13.6)", "Intercept     ", intercept
    print "(a,f13.6)",   "Slope         ", slope
    print "(a,f8.1,a)",  "Percentage fit", percent, "%"
    print "(/,a,f13.6)", "Mean X        ", xbar
    print "(a,f13.6)",   "Mean Y        ", ybar
    print "(/,a)", "ANALYSIS OF VARIANCE FOR REGRESSION"
    print "(/,a)", "Source of Variation   Sum of Squares   DoF     Mean Square"
    print "(a,f13.6,i6,f16.6)",   "Due to regression      ", ssdureg, 1, ssdureg
    print "(a,f13.6,i6,f16.6,a)", "About regression       ", ssabreg, nval - 2, meansq, &
                                  "    Variance"
```

Modern Fortran example code (4d)

```
! module reg_calc continued
    print "(a,f13.6,i6,f16.6,a)", "Total                  ", sumysq, nval - 1, fvalue, &
                                                            flabel
    ! Add means back to input data before calculating residuals
    xvals = xvals + xbar
    yvals = yvals + ybar
    print "(/,a)", "TABLE OF RESIDUALS"
    print "(/,a)", "Case No.     Y Value     Y Estimate       Residual"
    do i = 1, nval
      yest = intercept + slope * xvals(i)
      print "(i5,3f15.6)", i, yvals(i), yest, yvals(i) - yest
    end do
    deallocate(yvals, xvals, stat=dastat)
    if (dastat /= 0) then
      print "(/,a)", "Deallocating space for data failed"
    end if
  end subroutine calc_reg
end module reg_calc
```