bcs
The Chartered Institute for IT

Enabling the information society

Presentation to BCS Leicester Branch

8th November 2011

Fortran's Relevance in the 21st Century

Peter Crouch, Chairman BCS Fortran Specialist Group

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

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	Civil servant in the Department for Work and Pensions.
1993	Joined the British Computer Society
1997 – 2002	Chairman of BCS Birmingham Branch
2002 – 2011	Chairman of BCS Fortran Specialist Group

Early years of Fortran: 1954-1961

1954	**Development work starts in IBM**
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**

Pioneer Day Banquet, June 1982

Fortran pioneers gathered for a 25th anniversary banquet in 1982 include, from left, Richard Goldberg, Robert Nelson, Lois Habt, Roy Nutt, Irving Ziller, Sheldon Best, Harlan Herrick, John Backus and Peter Sheridan.

The first Fortran Standard 1962-1966

1962	Work on an ASA standard begins in order to promote consistency and enable portability
1962	IBM release Fortran IV for the 7030, 7090 and 7094; this removes some of the machine dependencies
1963	Most major US vendors have Fortran systems, either Fortran II or, increasingly, Fortran IV
1964	Technical work on the standard is complete. It is essentially a common subset of vendors' offerings
1965	ICT have Fortran IV on the 1900 and English Electric have Egtran on the KDF9
1966	The ASA/USASI standard is published *Clarifications are published in 1969 and 1971* *(The US standard is reproduced as an ISO standard in 1972)*

The 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.

FSG Activities 1970-1980

- FSG 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
- First contact with X3J3 members 1971
- FSG get on X3J3 mailing list and two-way flow of information and opinion starts
- FSG members attend occasional X3J3 meetings in the US
- 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 77: Principal changes

type CHARACTER

IMPLICIT, PARAMETER, SAVE

block IF

ENTRY

INQUIRE

new intrinsic functions

many detailed extensions to existing statements

real and double precision DO index

removal of Hollerith constants and data

MIL-STD-1753 1978: US DoD supplement to FORTRAN 77

IMPLICIT NONE

INCLUDE

END DO

DO WHILE

Additional functions for bit manipulation

FSG Activities 1981-1993

- FSG thrives, continuing in the same mode
- Fortran Forums are held in London (4) and Edinburgh (2), sometimes with visiting US speakers
- Some meetings are held outside London: Blacknest, Coventry, Jodrell Bank, Oxford, Reading, Rutherford Lab, Salford
- An experimental subgroup meeting is held in Glasgow as part of a drive for every Specialist Group to have meetings in Scotland
- Some FSG members become members of X3J3 and/or WG5 and attend regularly; others attend occasionally; the UK plays a significant part in development of Fortran 90

Development of Fortran 90

- **Originally scheduled for completion in 1982**
- **Renamed Fortran 8X, then Fortran 88 and finally completed in 1990 after rancorous discussions and attempts by some US vendors to derail the entire project**
- **Some US organizations attempt to retain Fortran 77 alongside Fortran 90**
- **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**

Fortran 90: Principal changes

- Free form source form
- Many minor modernizations and removal of arbitrary restrictions
- Whole array operations
- Facilities for modular data and procedure definitions
- Improved control over numerical computation
- Parameterized intrinsic types
- User-defined data types
- Pointers
- Some features deemed obsolescent – to allow for future deletion
- Still retaining compatibility for Fortran 77 programs!!

The F language: a regular subset of Fortran 90

- **Specified by a small group of experts involved in the development of Fortran 90 as a language that was highly regular in syntax and safe in use**

- **Intended to be easy to learn and reliable to use while retaining the powerful numerical features of Fortran 90**

- **Comprises the modern features introduced in Fortran 90 without the older, unsafe features of earlier versions of the language**

- **Full description given in "*The F programming language*", M.Metcalf & J.Reid, OUP, 1996, ISBN 0 19 850026 2**

Fortran 95: Principal changes

Part 1:

- FORALL
- PURE and ELEMENTAL procedures
- Initialization for pointers and for structures
- Designation of some older, duplicated features as ‘obsolescent’
- *Deletion of REAL and DP DO variables, PAUSE, ASSIGN and assigned GO TO, H edit descriptor*

Part 2:

- Varying length strings

FSG Activities 1993-2001

- Attendances wilt with the advent of the internet
- The FSG debates winding itself up but decides against
- In 1994, taking advantage of X3J3 and WG5 meetings in the UK, well-attended forums are held in Edinburgh, London and Oxford – but:
- a nadir is reached when the 1995 AGM is postponed because of a rail strike and the 1996 AGM is postponed due to rooms being double-booked; not everyone gets to know in time
- It is decided to hold only annual meetings plus special events
- NAG hold very successful ‘Fortran Futures 96’ and ‘Fortran Futures 98’ conferences “*in association with the FSG”*

Fortran 2003: Principal changes

- Parameterized derived types
- Object oriented programming support
- I/O enhancements, including stream access and asynchronous transfers
- Support for IEEE arithmetic and exception handling
- Standardised interoperability with C
- Support for ‘international usage’
- ASSOCIATE construct
- Data manipulation enhancements: allocatable components, etc
- Procedure pointers
- Scoping enhancements
- Access to command line arguments, environmental variables

Fortran 2003: Implementation Problems

- **Fortran 2003 turns out to be far more difficult to implement than had been foreseen**
- **Cray release the first full compiler, for some of their hardware, in December 2009 - six years after completion of the technical definition**
- **In 2010 IBM are the second vendor to release a full compiler**
- **Some suppliers let it be known that they plan to implement all of Fortran 2003 only if explicitly required by their customers**

ISO/IEC 1539-1:2010 (Fortran 2008) published October 2010

Coarrays as an extension for parallel processing

Submodules to reduce compilation cascades

Enhancements to aid optimisation

Data enhancements including long integers, maximum array rank increased to 15, available kinds, hyperbolic and other functions

I/O enhancements including getting unique unit numbers, new edit descriptors

New BLOCK construct with declarations

Bit manipulation procedures

Execution of command shell commands

Fortran Standards summary

1966 ANSI standard X3.9-66 (FORTRAN 66) - first programming language standard - 39 pages

1978 ANSI X3.9-78 (FORTRAN 77) also published as ISO 1539:1980 - relatively minor revision - 243 pages

1991 ISO/IEC 1539:1991 (Fortran 90) - major revision 294 pages

1997 ISO/IEC 1539-1:1997 (Fortran 95) – minor revision 356 pages

2004 ISO/IEC 1539-1:2004 (Fortran 2003) - major revision 567 pages

2010 ISO/IEC 1539-1:2010 (Fortran 2008) - major revision 603 pages

Compiler support for Fortran 2003 & 2008 Standards

- Table first publised April 2007, revised every 3 months
- Section on Fortran 2008 features added August 2009
- Information on 8 compilers currently available

- ACM Fortran Forum magazine – latest version
- Fortranplus website – previous version

  www.fortranplus.co.uk/resources/fortran_2003_2008_compiler_support.pdf

- Information courtesy of Ian Chivers and Jane Sleightholme, www.fortranplus.co.uk

Future Fortran standardisation activity

Development of Technical Specifications, subsidiary standards, on

- Further Interoperability of Fortran with C
- Further Coarray Features

will take place over the next two to three years.

The contents of these Technical Specifications will, subject to any changes found to be necessary from their use, be incorporated into future versions of the Fortran standard.

FSG Activities 2002-2011

The FSG is revivified:

- In 2002 a Forum was held to discuss UK requirements for inclusion in Fortran 2003
- Also in 2002 a successful application was made to the BCS to support three FSG members (reduced to one member latterly) to attend WG5 meetings to help put the UK case on standards
- In 2007 a very successful full-day meeting was held with the CCS to mark the 50th anniversary of the release of the first Fortran compiler
- FSG members organized the 2007 WG5 meeting in BCS London offices and held a reception for WG5 members
- In 2010 a successful meeting was held with the Institute of Physics to mark the 40th anniversary of the Fortran SG

'Fifty Years of Fortran' meeting January 2007

Alex Stepanov, John Backus (1924-2007) and Paul McJones - February 2004

Some Fortran-related videos

- Dreamworks Supercomputing video

http://www.youtube.com/watch?v=TGSRvV9u32M

This video doesn't mention the software used on supercomputers but most of the applications are written in Fortran.

- Two IBM Fortran films, from 1958 and 1982

http://www.softwarepreservation.org/projects/FORTRAN/video

Some current application areas for Fortran

- Weather forecasting and climate prediction
- Analysis of seismic data for oil and gas exploration
- Nuclear test ban verification
- Financial analysis
- Vehicle crash simulation
- Analysis of data from space probes and satellites
- Modelling of nuclear weapons
- Computational fluid dynamics

NEC SX-8 supercomputer as used by UK Met Office

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


EXCITING IT JOBS

THE 1,000 MPH CLUB

In an unremarkable business unit next to the River Avon in Bristol, an amazing project is being developed and IT is at its heart. Henry Tucker investigates.

Richard Noble is a man who likes cars that go fast, very fast. In 1983 he broke the world land speed record in his car Thrust2 when it reached 650.88mph, a record he held until September 1997 when his follow up car, Thrust SSC, broke the sound barrier to reach 763mph.

Despite still holding the record along with driver Andy Green, Richard is determined to go even faster and this time build a car that will go an improbable 1,000mph. This is the Bloodhound SSC (supersonic car), which in 2012 will be taken to the Hakskeen Pan in the Northern Cape Province region of South Africa for the attempt.

Before then though is the challenge of designing and building not only the car, but also the interest of children of all ages. This, along with reaching 1,000mph, is as important for Richard and his team.

06 ITNOW September 2010

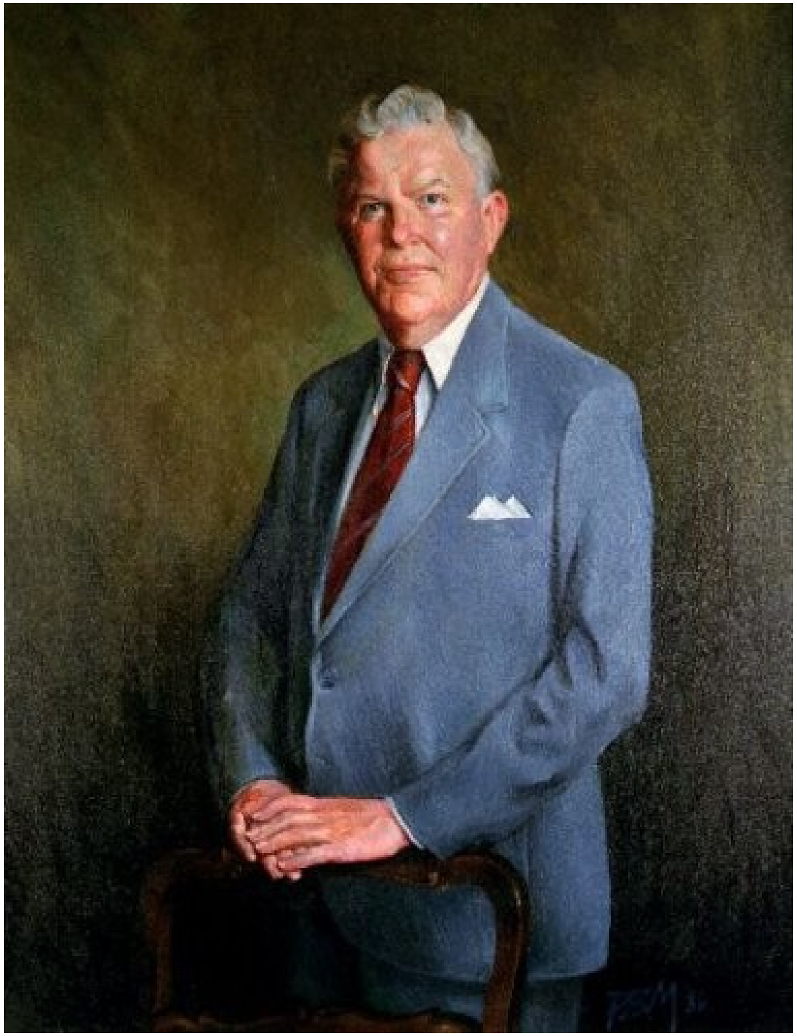
September 2010 ITNOW 07

Harvard Lomax, 1922 – 1999

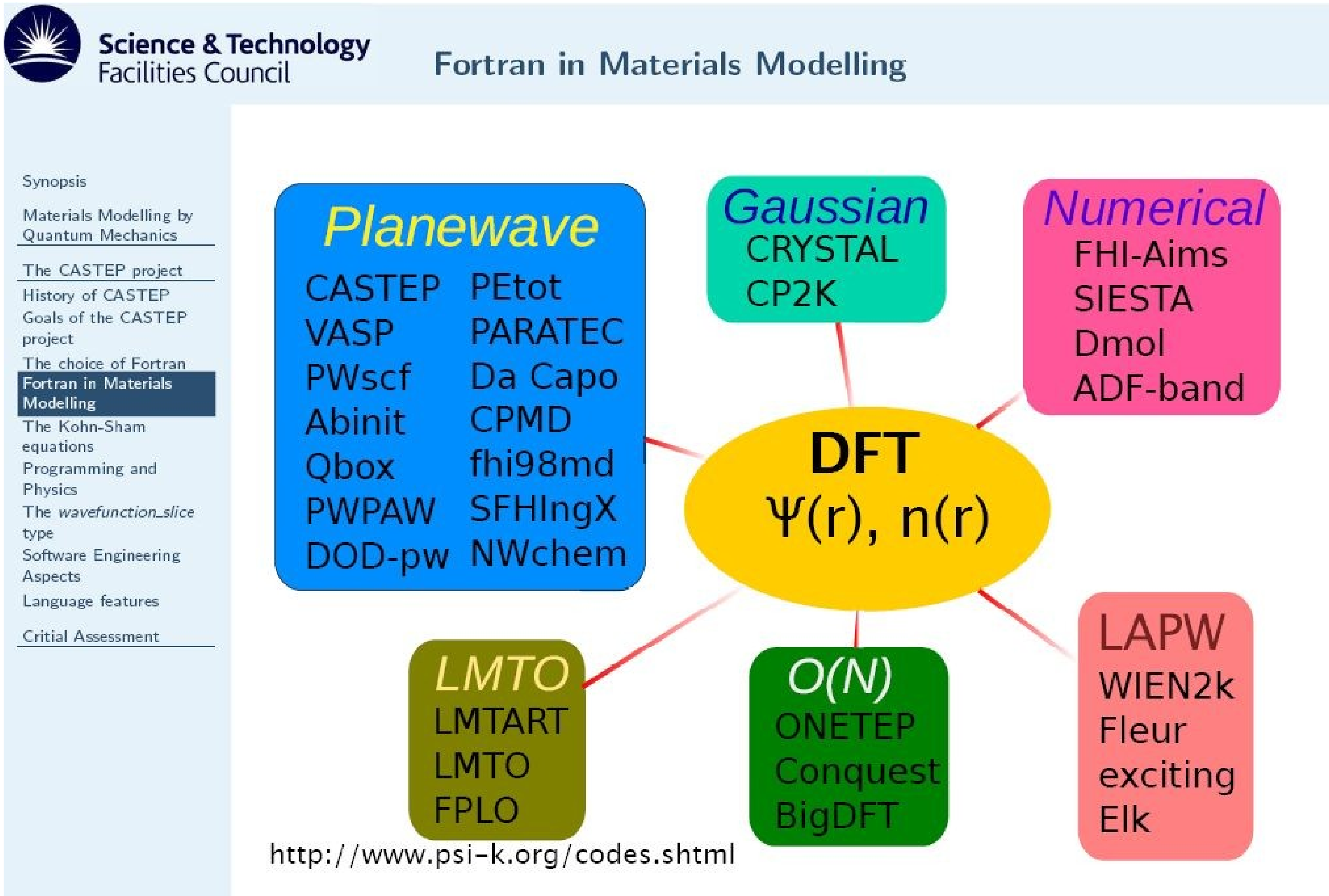
The father of “The Numerical Windtunnel”, CFD

NASA Ames Research Center, Moffett Field, California,

1944 – 1994



The CASTEP project:
Materials Modelling by Quantum Mechanics

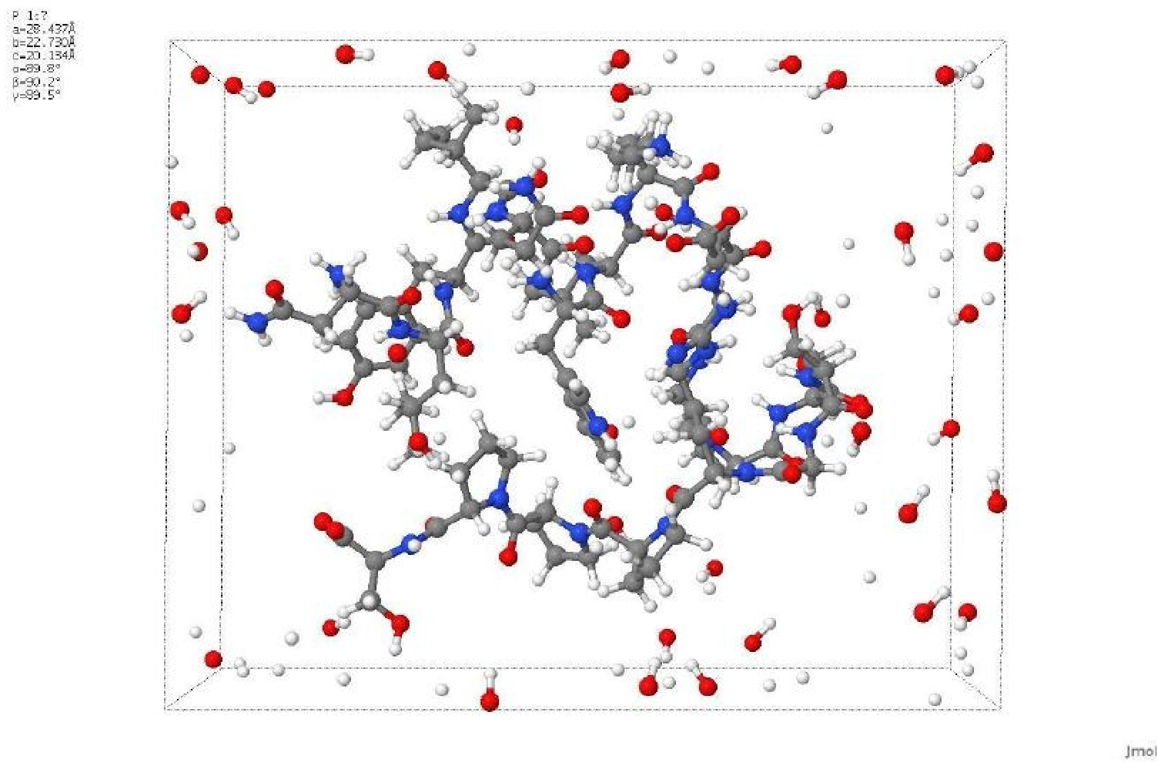


CASTOR HPC Capability

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.

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, Autumn 2011

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

Many thanks to David Muxworthy for his presentation to the joint BCS/IoP meeting in June 2010 marking the 40th anniversary of the BCS Fortran Specialist Group. Many of the slides in this presentation are taken from it.

Other slides are based on parts of presentations given by Keith Refson and Ian Chivers and Jane Sleightholme following the FSG AGM in September 2010.

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.

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

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
```