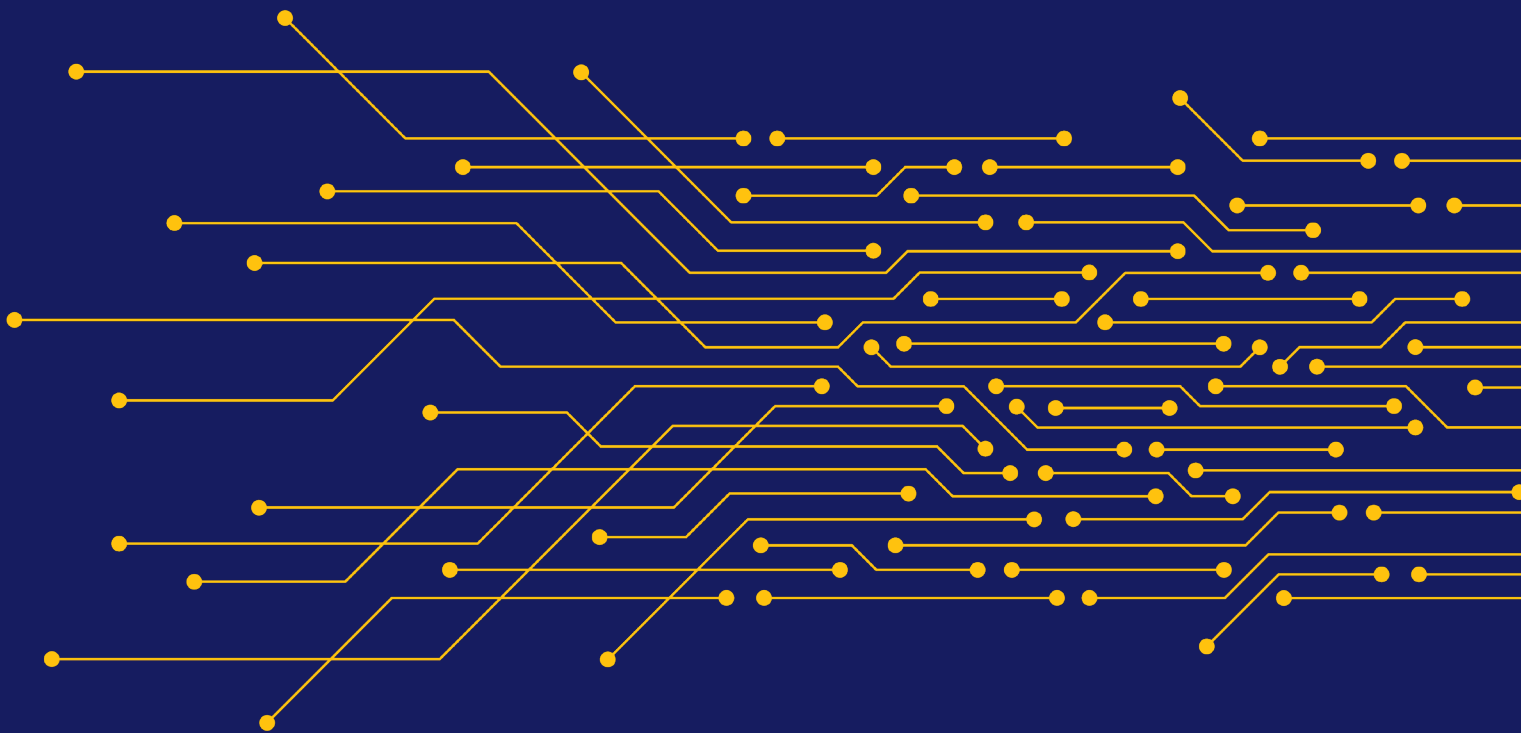


# The world's oldest mathematical software library

*STFC Scientific Computing manages the Harwell Subroutine Library (HSL), the world's oldest mathematical software library. It has been in continuous development and use for over 60 years.*

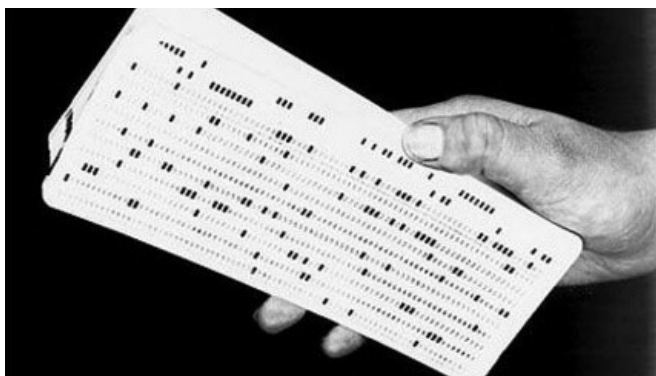


This mathematical software library was born from the need to help nuclear physicists and associated researchers with coding problems, at the original Harwell campus in the early 1960s. Six decades on, it is still providing easy access to complex codes and software packages that enable faster, more efficient scientific research to this day.

## How it began

After the RAF Harwell airfield was closed in 1950, the site became home to the Atomic Energy Research Establishment. This branch of the Ministry of Defence was responsible for researching the applications of nuclear energy, and was home to a broad host of physicists, engineers, and mathematicians, all working on state-of-the-art equipment.

This included 'Stretch', one of the first transistorized computers developed by IBM, which the scientists would use through Fortran, a programming language designed for scientific computing. At the cutting-edge for its time, Stretch was the largest, fastest, general purpose computer in the world – one of the original 'supercomputers'. It could take a whole day to run one calculation, so researchers needed to check and double-check their work, as a coding error could cost them the whole day of calculation time. Mathematical scientists would help check and recheck the, sometimes rudimentary, codes being produced to run on Stretch.



One of them was mathematician Michael J.D Powell, who joined the team at Harwell after graduating from the University of Cambridge. Powell was an expert in the fields of numerical approximation and optimisation and in 1963 he proposed the idea of creating a collection of subroutines (or codes) designed specifically for scientific computing. They named this project the 'Harwell Subroutine Library', which came into being in May 1963.

## Explaining Subroutines



Subroutines are small blocks of code in a programme which perform a specific task, and the Numerical Analysis Group (renamed the Computational Mathematics Group within STFC Scientific Computing in 2017) compiled a huge library of subroutines in a wide range of computational disciplines. Professor Iain Duff, former Group Leader and now an Honorary Scientist in the Computational Mathematics Group, was a DPhil student at Harwell in the 1970s and one of the mathematicians brought in to develop numerical algorithms.

“People were making their own codes which then didn't work – it wasn't their area of expertise – so we were called in to help. This was essentially why Mike Powell made a proposal for the library, to establish something that was available to everyone and would be of a high numerical standard.

“Back then, everything was done using stacks of computer cards which the scientists would take by car to the IBM Stretch machine at Aldermaston. They would do a computer run and bring back the printed output by about lunchtime, and then take the next lot of cards for the afternoon run. So, you usually got two computer runs each day. The calculations that used to take at least half a day in the 1960s and 70s can now be done in a millisecond – that's one thousandth of a second – on a modern smartphone!”

***Professor Iain Duff, former Group Leader, Honorary Scientist  
in the Computational Mathematics Group***

HSL today still includes some of those original Fortran subroutines written by Powell and his colleagues, and over the years the library has evolved and has been extensively used worldwide on a wide range of hardware, from massively parallel supercomputers to modern PCs.

In recent years, one of the original codes developed by Mike Powell was used to optimise a plug-in for NASA's General Mission Analysis Tool and efficiently map the trajectory for one of the NASA missions.

The library holds subroutines and software packages created by the Computational Mathematics Group and their academic partners, mostly written in Fortran.

The original codes were designed so that they could be used to solve multiple problems rather than being targeted at a particular application. This is still the case today and they can be just 'plugged in' as a general solver, although they do target linear algebra and, in particular, sparse matrices.

These are matrices which have lots of zeros in them as there is no direct interaction between all of the components. So, for example, each bead in a necklace is only connected to the adjacent beads, forming a string. The beads are not directly interacting with all of the other beads. When modelled numerically this would be an example of a sparse matrix.



**“For a matrix with large-scale symmetric indefinite systems – where there are vast numbers of zeros not interacting with the numbered components – our codes perform better than virtually any others. This is where the HSL codes shine!**

**“Solving these systems is key to projects such as making self-driving cars work safely and efficiently, and we’ve been getting some interest recently from companies working in this space.”**

***Dr Tyrone Rees, Computational Mathematics Group Leader***

Currently, the subroutines and packages created and located within the library are available under licence from STFC. The archive, located within the HSL website, contains subroutines written as far back as the library's inception in 1963. All the subroutines include a specification sheet, which explains the variable names, what they do, and instructions on how to interface these subroutines to your work.

The most popular licence option is Coin-HSL, which is a bundle of 5 linear solver packages designed for optimisation. Any users of the MATLAB platform will also be familiar with certain HSL licensed codes as the MATLAB creators, MathWorks, own a license for the linear solver MA57, which is incorporated into MATLAB. For more information, to read the codes or to acquire a licence, you can access the library at [hsl.rl.ac.uk](https://hsl.rl.ac.uk)

Professor Jennifer Scott, who was formerly the Computational Mathematics Group Leader and whose time is now split between STFC and her academic research at the University of Reading, has produced a history of the Harwell Subroutine Library. It gives detailed information about how the HSL was created and how it has evolved over the years.

Find out more: <https://bit.ly/3wGMZ7C>

*Written by Adam Grady and Marion O'Sullivan  
Designed by Dascha Choudhury-Gibbons*



Science and  
Technology  
Facilities Council

# Scientific Computing

