Science and Technology Facilities Council

Technical Report
STFC-TR-2023-002

HSL@60: a brief history of the HSL mathematical software library

J Scott

May 2023

Enquiries concerning this report should be addressed to:

RAL Library
STFC Rutherford Appleton Laboratory
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445577
email: library@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: https://epubs.stfc.ac.uk

Accessibility: a Microsoft Word version of this document (for use with assistive technology) may be available on request.

DOI: 10.5286/stfctr.2023002

ISSN 2753-5797

STFC Author Identifiers (ORCIDs)

Author ORCIDs are provided where available.

Jennifer Scott	0000-0003-2130-1091

HSL@60
A BRIEF HISTORY OF THE HSL MATHEMATICAL SOFTWARE LIBRARY

JENNIFER SCOTT*

Abstract. The HSL Mathematical Software Library (https://www.hsl.rl.ac.uk/) started life in 1963 as the Harwell Subroutine Library. It is the world's oldest mathematical software library to have been in continual use. Over the years, the HSL Library has continually evolved and has been developed and maintained by a small research group of numerical analysts that has been at the Rutherford Appleton Laboratory since 1990. The group's interests lie in the development of numerical algorithms and their underlying theory and then implementing these algorithms in state-of-the-art software. The focus for many years has been on sparse problems. The HSL Library has been extensively used on a wide range of computing platforms, from supercomputers to modern PCs and laptops. The current version of the HSL Library is still widely used and it remains a highly respected source of software for solving sparse problems. This report celebrates the 60th anniversary by presenting a short historic overview, summarising the key milestones of the HSL Library 1963–2023.

1. The birth of the Library. The Atomic Energy Research Establishment (AERE) at Harwell was founded after the end of the Second World War in 1946 on the former RAF Harwell site, sixteen miles south of Oxford. It was the UK's first Atomic Energy Research Establishment and was initially under the Ministry of Supply. The main objectives were to tackle the postwar energy crisis and advance nuclear technology. The first nuclear reactor in the UK, a small research reactor known as GLEEP, went critical at Harwell on 15 August 1947. AERE Harwell expanded rapidly and, in 1954, it was incorporated into the newly formed United Kingdom Atomic Energy Authority (UKAEA). During the 1950's, the scientific computing needs of the UK were dominated by nuclear research. The Atomic Weapons Research Establishment (AWRE) at Aldermaston removed all nuclear weapons research from AERE and it became the main computer centre, initially running a Ferranti Mark 1 machine, later supplemented by an IBM 704 and an IBM 7090 in 1958.[1] Most of the computing at Harwell in the early 1960s was performed on the machines at Aldermaston and at Risley. Jobs were physically transferred by magnetic tapes in a van, with a satellite IBM at Harwell reading cards onto tape and printing results from tape. Scientists did their own programming, mainly using the Fortran programming language, which was originally developed by IBM in the 1950s for scientific and engineering applications. The first Fortran compiler was released in 1957 and by 1963 there were over 40 Fortran compilers in the world for different computers. The 1966 standard for Fortran was the first ever for a computer language; it helped enable portable software to be developed.

The *Preparation of Programs for an Electronic Digital Computer* (sometimes called WWG from the initials of its authors Wilkes, Wheeler, and Gill) is regarded as the first conventionally published book on computer programming [5]. It was written for the EDSAC (Electronic Delay Storage Automatic Calculator) computer that began operation at the Mathematical Laboratory of the University of Cambridge in 1949 as the world's first regularly operated stored program computer. Initially written as a report in Cambridge, the book was published in 1951 by Addison-Wesley Press in the US. It was the first book to describe a number of important concepts in programming, including the first account of a library of reusable code that aimed to provide building blocks for an array of programmes. The library consisted of around 100 subroutines implementing mathematical operations such as the calculation of trigonometric functions and arithmetic operations on complex numbers. The library was a physical collection of paper tapes that were stored in a metal filing cabinet. This included a "library catalog" that described how to use each subroutine. One of the key drivers of the library was to reduce the errors in programs. At the time WWG that was published, very few digital computers existed and demand for the book was so limited initially that it took six years for the first edition to sell out; the second edition appeared in 1957 [31] [2]. WWG arguably became one of the most influential textbooks of the early computing era. Today,

*STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX, UK
Correspondence to jennifer.scott@stfc.ac.uk.

[1] https://www.chilton-computing.org.uk/

[2] A facsimile of the second edition is available at https://www.computerhistory.org/collections/catalog/B286.70

a seller of rare books has a first edition costing close to £1000[3]!

Initial discussions on establishing a mathematics software library at AERE Harwell began in January 1963. Notes by Curtis ahead of the first meeting explained: *The advantages of establishing and maintaining a proper library of standard programs and subroutines for a computer are widely recognised to be enormous. To do so, however, involves considerable effort on the part of its organiser, and it is important that the work be well done* [7]. Powell goes on to state that the purposes of a software library at Harwell are:

(i) to have available efficient programmes which perform common algebraic manipulations and which evaluate common functions, and

(ii) to keep copies of programmes whose writers think may be of use to somebody else sometime.

The reasons for having this library are:

(a) to avoid duplication in writing programmes,

(b) to have routines which are probably more efficient than those which the average user writes for a one-off job, and

(c) to speed up the programming of routine calculations.

Following the initial meeting, much of the work of the numerical analysts at Harwell (led by Powell) became directed towards providing users with good Fortran subroutines for their mathematical calculations: the Harwell Subroutine Library was born. Initially, it was for internal use on an IBM 7030 (STRETCH) machine; in 1967 it was converted to run on an IBM 360.

2. The Library at Harwell (1963–1990). By the end of 1963, about 90 subroutines were available in the Library; two years later, this had increased to around 150 subroutines. They encompassed software for rational approximation, special functions, two-point boundary-value problems, Gaussian quadrature, linear least squares, linear programming, random number generators, sorting, linear equations, eigenvalues, polynomial fitting and optimization. Most routines were written in Fortran by contributors at Harwell, but some were from the nearby Culham Laboratory and the Atlas Computer Laboratory (which in 1975 merged with the Rutherford High Energy Laboratory to become the Rutherford Laboratory and then, in 1979, merged with the Appleton Laboratory to became the Rutherford Appleton Laboratory).

With a growing reputation, from 1964 onwards the Library was distributed externally upon request. The first Library catalogue in report form was issued in April 1966. Thereafter, a computer generated list of routines was made available until a revised catalogue was released in August 1971 [22]. In the early 1970s, the intention was to re-issue the catalogue annually, with a Harwell Subroutine Library Bulletin coming out approximately every 4 months to publicise new routines as well as deletions. The introduction to the catalogue explains that *It stands as a precise definition of the library and serves as a reference document to be used by users of the Harwell computer. New users to the Harwell computer will find the special section on how to use the library useful as an introduction to the library facilities.* In practice, producing a new catalogue each year was found to be too ambitious. Instead, the aim changed to releases being a minimum of two years apart, with supplements to the catalogue used to keep things up to date. For example, supplements to the 1973 catalogue [23] were released as AERE reports in 1974 and 1977 [24, 26]. The latter explains that *Many of our new subroutines have been written with portability in mind and it is the intention of some of our authors to write code that conforms to the Fortran IV ANSI standard (1966).*

The organization of the Library was established by the time of the first catalogue and, in many ways, the design has remained substantially unchanged. The Library is split into chapters, each identified by two letters. For example,

- `MA`: matrix routines (solvers)
- `MC`: matrix routines (manipulation)
- `EB`: unsymmetric eigensystems
- `VC`: data fitting by polynomials and spline functions

[3] `https://www.peterharrington.co.uk` (April 2023)

Within each chapter, each subroutine has a 2-digit identifier, generally allocated chronologically, for example (from the 1973 catalogue):

- `MA07`: solves a banded structured system of equations
- `MA12`: solves an upper Hessenberg system of equations.

The limit of 6-character names imposed by all early versions of Fortran restricted the ability to give the routines more easily recognisable names (this was also an issue for other contemporary mathematical software libraries). Write-ups for individual routines (or subroutine specification sheets as they became known) that explain how to use a subroutine were available from filing cabinets in the computer reception area at AERE Harwell; alternatively, those without access to this area could make a request to have a copy sent to them by post. The format of the specification sheets was standardized and, again, in broad terms, remains substantially unchanged.

Some routines perform major tasks, such as solving a sparse linear system; these are the ones that external users are most likely to want. Other routines perform what may be regarded as subsidiary tasks, such as scaling or ordering a sparse matrix. These are required by the sparse solvers and so by developing them as independent routines (and not as part of a solver), duplication of effort is avoided. A modular approach also helps with software testing and maintenance.

By the 1970s, an appreciation of the work involved in establishing and maintaining numerical libraries was generally understood and the new subject of mathematical software lying between numerical analysis and computer science grew at a tremendous pace. At this time, a number of other important software library projects were getting going elsewhere, including the Boeing BCSLIB library in the US and the Numerical Algorithms Group (NAG) library in the UK. The latter started in 1970 as a universities' project but became a not-for-profit company in 1976. Also in 1970, the commercial International Mathematical and Statistical Libraries (IMSL) was incorporated. As well as these general-purpose libraries, specialised libraries began to be developed, including the eigenvalue package EISPACK (1976) and the linear equations package LINPACK (1979), which were later largely superseded by LAPACK (1992). A useful early reference on the emergence of subroutines libraries up until the end of the 1970s is given by Fox, Hall and Schryer [16].

Before 1966, most of the numerical methods implemented within the Harwell library were taken from the existing literature. Once the Library was established, there was a move to spend more time on developing new numerical methods along with underlying theory. User's views were sought as to what would be particularly useful but generally new areas were chosen by the Numerical Analysis Group at Harwell. The Group was initially part of the Theoretical Physics Division; it joined the newly-formed Computer Science and Systems Division in 1973. At that time, led by Group members Curtis, Reid and Duff, interest in sparse matrix calculations rapidly developed and Harwell became recognised as a centre of excellence in the field. The first sparse linear system solvers were `MA17` and `MA18` in 1971 [8]. The most important as well as the most enduring sparse linear system solvers from the late 1970s were the multifrontal package `MA27` for sparse symmetric systems (positive definite and indefinite problems) [14, 10] and `MA28` for general sparse systems [13]; both were ground-breaking and, although superseded by more recent packages, they remain in use today.

The "V" optimization chapter also grew very rapidly during these Harwell days, driven in part by the requirements of on-site physicists, but also because of the friendly rivalry between Group members Powell and Fletcher. At one stage in 1970, a new optimization routine appeared every month, seemingly alternating between the pair of them.

Computing facilities at Harwell in the 1980s were among the most advanced available, with a Cray-1 installed in 1981, and upgraded to a Cray-1S in 1982. This was followed by a Cray X-MP in 1986 and, finally, a Cray-2 in 1987 at the then cost of £13 million. At the time, the Cray-2 was the world's most powerful supercomputer, with a peak performance of 1.9 gigaflops. Some HSL routines were instrumental in the benchmarking and to some extent the purchase of these machines, particularly the CRAY-2.

Library documentation and TSSD
A very important part of the Library is its documentation. Early on it was recognised that user documentation needed to be held in a machine readable form that could be updated and edited easily. It also had to be able to incorporate mathematics (including equations, matrices, tables etc) in a straightforward way. Previously, mathematical expressions had to be laboriously typeset by typists using special typewriters that did not produce high quality layouts and were not capable of automation. To overcome this, a sophisticated computer typesetting system called TSSD (which stands for Typesetting System for Scientific Documents) was developed in the mid 1970s at Harwell by Numerical Analysis Group member Hopper [25]. TSSD was specifically intended for the Harwell Subroutine Library documentation but it was also made available to other potential users at Harwell and later to external organisations (including NAG Ltd). TSSD was written in IBM Fortran IV and originally ran on the Harwell IBM 370/168. It revolutionised both the quality and the maintenance of the Library specification sheets and the catalogue.

TSSD predated the TeX typesetting system, which was initially released in 1978. In 1985, LaTeX was created in the US by Lamport as an addition TeX with the objective of making it easier to produce documents within TeX. TSSD is arguably as easy to use as LaTeX and TSSD documents are of the same quality as LaTeX ones. However, TSSD did not receive the same exposure or take up as LaTeX and the Library documentation gradually moved to using LaTeX. By 2006, all new HSL specification sheets (and major revisions to older specification sheets) were typeset using LaTeX. Nevertheless, because the effort involved made it impractical to convert all the existing documentation from TSSD to LaTeX, many of the older packages that are still available retain their specification sheets in TSSD and Hopper (who is now 90) has continued to provide updates to TSSD that have enabled it to be run on our current operating systems.

3. The transition to RAL (1990s). By the late 1980s, the requirements for nuclear research and development had reduced significantly, with the last three of the fourteen experimental reactors that were built on the Harwell site ceasing to operate in 1990. The number of AERE employees fell to 2000, down from 6200 in 1959 and 4500 in 1973 [18, 27]. The site moved into non-nuclear and commercial research, and then to reactor decommissioning, waste management and site restoration. In 1996, all non-fusion research of the UKAEA was privatised as AEA Technology plc. Its successor as a quoted company, AEA Technology Group plc, eventually went into administration in 2012.

As the UK political climate and research priorities changed, it became clear that the Numerical Analysis Group no longer had a bright future as a research group at Harwell: to survive meant either relocating or seeking to work on commercially profitable projects. After extensive high-level negotiations, an agreement was reached that in April 1990, the core team of numerical analysis researchers (Duff, Gould, Reid and Scott) would move to the Central Computing Department (CCD) of the Rutherford Appleton Laboratory (RAL). RAL is also located on the old RAF airfield. In 1990, it was the responsibility of the Science and Engineering Research Council (SERC), the forerunner of the Engineering and Physical Sciences Research Council, which was established in 1994. The CCD was based in the Atlas Centre and was responsible for mainframe computing and networking activities at RAL. Mainframe services were provided on IBM compatible computers into the 1990s, with a Cray X-MP/48 (which at that time was the most powerful supercomputer on the market) installed in 1987 and replaced in 1992 by a Cray Y-MP. RAL's role as a provider of services for the Research Councils' high performance computing programme came to an end in 1999.

The relocation of the Library's main researchers to RAL necessitated a legal agreement between AERE Harwell and SERC. At the time, AERE Harwell and then AEA Technology owned the copyright on all the routines in the Harwell Subroutine Library. After 1990, new routines were copyrighted to SERC and its successors. AEA Technology (and subsequently Hyprotech Ltd, a wholly-owned subsidiary that later became a part of the American-owned company Aspen Technology) took on responsibility for marketing and distributing the Library. Their interests were the potential to generate revenue from the software while allowing ex-Harwell researchers within their organisation to retain full access to the whole Library.

Around the time of the Group's move to RAL, it was becoming clear that subroutine-based interfaces to optimization software were increasingly inconvenient for users. The size of the problems that could be solved was growing but the requirement to specify nonlinear functions and derivative structures via subroutines was both tedious and error prone. The first Group-led project to mesh both an optimization solver with an external "modelling environment" was the Fortran 77 package LANCELOT [6] and although there was a subroutine interface described in a traditional HSL style specification document, that was not how it was used in practice. Thus, LANCELOT was out of place within HSL and the decision was made that it should be a separate software project supported and maintained by the Group.

In the late 1980s, there was interest from NAG Ltd in the Harwell sparse matrix routines, as they complemented the routines within the main NAG Library. In 1988, Release 1 of the Harwell Sparse Matrix Library (HSML) was marketed and distributed by NAG Ltd. It contained 19 packages. The original intention was that HSML would not include the newest packages but it would be one release behind the main Harwell Library. Release 2 of HSML with 45 packages appeared in 1995. However, despite additional talks with NAG in 2000, there were no further releases. This was partly because each release involved significant work for the Numerical Analysis Group and, disappointingly, the expected increase in users and exposure of the Harwell subroutines did not materialise. The main benefit of HSML for the Group was the opportunity to gain experience from NAG about their software engineering practices; this led to better-defined development and maintenance procedures [20, 29].

Release 10 of the Library in 1990 came with the first publication, in book form, of the whole collection of specification sheets. These were sent to users/organisations that purchased copies of the whole Library. For Release 11 in July 1993 and Release 12 in December 1995, the specification sheets were printed in two volumes. These were both expensive and cumbersome. Buying a set of the documentation cost £95 in 1995. With the rapid growth of electronic documents, no further volumes were printed (and many copies of the 1995 documentation ended up being scrapped!).

In the mid 1990s, the Rutherford Group still comprised only the four members who had come from Harwell and thus there was limited time for working on software. However, the Library greatly benefited from collaborations with French PhD students who were supervised by Duff as part of his part-time role at CERFACS in Toulouse[4]. In 1994, the main focus of the UK Research Councils' national supercomputing programme moved to the Edinburgh Parallel Computing Centre, resulting in job losses in the Atlas Centre and pressure on funding for the Group increased. From 1996 onwards, the Group became heavily dependent on applying to EPSRC for research funding. Impact was not a criteria used by EPSRC in the 1990s when assessing applications for funding from the Mathematics Programme and this meant focusing grants on the research aspects of the Group's work, rather than on the software side. Over the years, this has changed significantly, with the impact and the wider applicability of work funded by EPSRC becoming much more important and high quality software is now appreciated and encouraged as a key outcome.

Significant packages for sparse linear systems that were developed in the 1990s included `MA48` (1993), which superseded the earlier LU factorization code `MA28`; `MA38` (1995), which was later developed further externally and became UMFPACK [9]; and `MA41` (1995), which was one of the origins of the modern MUMPS package [2, 28]. `MA38` and `MA41` as well as a sparse QR package `MA49` (1998) were co-authored by CERFACS PhD students. In addition to the solvers, there were important routines for sparse matrix orderings and scalings; these are key to the success of the sparse solvers. They were developed as separate packages that could easily be incorporated into the HSL solvers and used elsewhere by external users and other packages. Important examples include the approximate minimum degree ordering algorithm of `MC47` [1] and the matching-based scaling and ordering of `MC64` [12].

The mid 1990s saw the first Fortran 90 compilers become available and the first Fortran 90 routines (most notably the sparse linear solver `HSL_MA42` [15] for systems of unassembled finite elements) were included in Release 12 (1995). The naming convention was extended so that packages in Fortran 90 (or later versions) are prefixed by `HSL_`. Solvers from this period onwards were designed to incorporate the use of BLAS (Basic Linear Algebra Subroutines) and LAPACK routines. Many vendors of compilers and

[4] `https://cerfacs.fr/en/`

operating systems provide precompiled and optimized libraries for these dense linear algebra subroutines; this can significantly enhance performance of the solvers across a wide range of platforms.

4. HSL 2000 onwards. Although the Library remained a general-purpose collection of mathematical software until the end of the century, it was too large for a small research group to grow, develop and maintain. It became increasingly clear that its focus needed to shift towards becoming a specialist library, with its sparse matrix routines at its heart. Such a library would complement and exist alongside other (external) software libraries.

By 2000, the Library was used by more than 1500 organizations worldwide and packages had been incorporated under licence into over 60 commercially available products. However, the many disruptive changes within AEA Technology during the 1990s were not beneficial for the Group or the Library. With yet another change of the personnel at AEA Technology who had responsibility for the Library, it was recognised that the original 1991 agreement needed to be revised and new licences drawn up for Library users. It was decided that the Library should be split into two disjoint parts: HSL 2000 and HSL Archive. The latter would comprise older packages that the Group no longer wanted to support or develop, or had been superseded, either by packages in HSL 2000 or routines in other freely-available libraries, most notably LAPACK. HSL 2000 and HSL Archive saw the first use of the internet for the distribution of the software (although CD-ROMs continued to be used for shipping the complete HSL 2000).

Not owning the copyright of the pre-1990 routines and the relationship with AEA Technology limited the Group's freedom to distribute the Library. Although there were reduced rates for academic users, it was not until HSL 2000 that agreement was reached that individual packages could be made available without cost to academic users for research purposes. This change came about in part because at that time the Group was being funded through a large EPSRC grant and EPSRC was rightly keen that others should benefit from the outcomes of its funding. Initially, only UK academics were offered free access; this was extended to academics worldwide in 2010. The only requirement became having a valid academic email address in a country not on any UK government list of banned countries. HSL Archive has always been freely-available for non-commercial use. While it is not possible to support and maintain HSL Archive to the same standards as as the main Library, it is not a static collection as HSL routines are moved into it as the Group and its interests, as well as external software, evolve.

The widening of academic access came as part of a lengthy renegotiation of the contract between STFC and Aspen Technology. At the time of the new agreement, Aspen Technology was concerned about protecting their interests in HSL and, in particular, wished to prevent their potential competitors from obtaining access to HSL routines. These competitors are included on a list of companies that HSL packages cannot be licenced to without prior permission from Aspen Technology. The agreement of 2 June 2010 is still in place and, although Aspen Technology has had no interest for many years in marketing or otherwise promoting and selling HSL, their payment of an annual fee has been invaluable in helping to fund staff time for the ongoing maintenance of the Library.

One of the most important packages in HSL 2000 was `MA57` [11], which was designed to supersede the sparse symmetric linear solver `MA27`. `MA57` was later to become one of the packages behind the Matlab backslash function, thus hugely increasing its usage (although without most Matlab users being aware of using an HSL package). A key difference between `MA57` and `MA27` is that the former makes extensive use of level 3 BLAS, which has the potential to significantly enhance performance and permit a degree of parallelism if multithreaded BLAS are used. HSL 2000 included the first MPI routines. These were coarse-grained parallel solvers that were built on top of existing solvers. For instance, `HSL_MP42` was an MPI version of `MA42` (`MP` was chosen to distinguish MPI routines).

HSL 2002 was the first version of the Library to be fully thread-safe. With the move towards distribution via the internet and the greater flexibility that this offered, it was decided after HSL 2002 that new packages should be made available on completion, and not to wait until the next formal HSL release. Formal releases were made in 2004, 2007, 2011 and 2013. After 2013, HSL moved solely to a rolling release schedule, with new and upgraded packages being added as available.

Major new parallel (OpenMP) sparse solvers `HSL_MA87` [19] and `HSL_MA97` [21] were released in 2009 and 2011, respectively. More recently, with the ever-increasing importance of preconditioned iterative solvers for large sparse systems, the Library has included a limited number of general-purpose algebraic preconditioners for use with iterative solvers.

On the optimization side, the Fortran 95 package LANCELOT B was released in 2002. Work was already underway on the more ambitious GALAHAD library [17] that mirrored HSL's integrated component-based design, but again with modelling environments as a focus. As the HSL optimization software had mostly been designed to solve small problems (which was all that was feasible at the time), it did not fit into this model, and by the 2011 Release of HSL, all the optimization packages had either been moved into the HSL Archive or were flagged as having been superseded by GALAHAD. GALAHAD currently comprises over 100 integrated packages, and offers interfaces to C, Matlab, Python and, very recently, Julia. GALAHAD is an important user of the HSL packages for solving sparse symmetric linear systems and it has influenced the design of and features included in these packages.

HSL software continues to be written in Fortran, with the version of modern Fortran used being dependent on the widespread availability of reliable compilers. Matlab and C interfaces for some key routines were included for the first time in HSL 2011. These widened the potential user base; in particular, the Matlab interfaces facilitated the use of HSL for teaching purposes (typically masters courses) and their use by many researchers and PhD students who have no background in languages such as Fortran and C. One of the most widely-used Matlab interfaces is for the algebraic multigrid solver `HSL_MI20` [3]. In the last few years, the number of packages offering Matlab, C and/or Python interfaces has grown, to meet the requirements of today's users of the Library.

5. 2023 and beyond. The Library has constantly evolved over the last 60 years, reflecting developments in numerical analysis and the research interests of the Group. Of course, there have been many changes within the Group, with new members joining and staying for varying lengths of time (ranging from a few months to many years). They have worked with the original Rutherford Group to make significant contributions to the Library. New developments have also been driven by the constantly evolving computer architectures and by research grants awarded to the Group. Today, HSL is a specialist collection of packages for large-scale scientific computation. It is known for having a high standard of reliability and has an enviable international reputation as a source of robust and efficient numerical software, and is particularly renowned for its software for the solution of sparse systems of equations. This is designed to be general-purpose (rather than driven by a specific application). It is used worldwide to contribute to solving a huge variety of problems, primarily by academics but also by a number of commercial companies. Recent non-academic users are extremely diverse and include the leading Formula 1 development teams, a research company working in the field of robotics, a family-owned European food producer, a global data analytics company that focuses on retail and consumer intelligence, and multinational chemical and software corporations.

It is important to note that the income generated by HSL licence sales has always been small. A one-off fee is charged for a commercial licence. The fee depends on whether the HSL software will be used in-house for research and development or incorporated into a third-party product that is sold or otherwise distributed. In the latter case, the amount charged takes into consideration the expected contribution of the HSL software and the sales that it may contribute to. The income from licences helps fund staff time for working on maintaining the software but it is not (and never has been) sufficient to cover any research costs. The Group provides limited user support (such as fixing reported bugs in the codes and helping with the choice of packages) but as a research group it lacks both the motivation and the manpower to offer consultancy services.

A large percentage of current users of the HSL sparse solvers (both academics and non academics) employ them within the Ipopt software library for large scale nonlinear optimization of continuous systems. Ipopt [30] is the de facto open-source standard for interior point optimization. It needs to solve a series of sparse symmetric indefinite linear systems and for this it relies on third-party code. Ipopt has interfaces to the HSL sparse solvers `MA27`, `MA57`, `HSL_MA77`, `HSL_MA86`, and `HSL_MA97`, with different solvers proving the best choice for different problems and architectures.

But what of the future? RAL is a part of the Harwell Science and Innovation Campus, a campus that is now home to over 6,000 engineers, scientists and innovators from 225 organisations, encompassing start-ups to biotech unicorns. Reflecting the changing times, the Numerical Analysis Group was renamed the Computational Mathematics Group in 2017. It currently has 8 permanent members and it is anticipated that a significant number of new appointments will be made in the next couple of years, diversifying the membership and the work that is undertaken, with a growing emphasis on joint projects involving researchers from other areas within STFC. This can be seen in some ways as coming full circle, back to the model of the early AERE Harwell days, where HSL was developed primarily for Harwell researchers and involved mathematicians and software engineers. The latter, with their complementary skills, are hugely important but the Group and HSL has generally lacked software engineers; it is very positive that this is now changing. It is fully expected that an emphasis on research that leads to the development of software for users within and beyond the Group will remain a key activity, along with supporting and maintaining the existing software while it continues to be in demand. In addition, further interfaces will be added to allow a wide community of users working in different computing environments to use the packages to solve diverse problems. In particular, Julia interfaces will become available in the near future.

Driven by the licence restrictions imposed on HSL by the agreement with Aspen Technology combined with the general move for research outcomes, including software, to be open access, alternatives to HSL have already been considered. The Group's Sparse Parallel Robust Algorithms Library (SPRAL) is a small open-source library that began in 2015 for sparse linear algebra and associated algorithms[5]. Its key package is SSIDS (Sparse Symmetric Indefinite Direct Solver). It was initially implemented for parallelization over NVIDIA GPUs but now enables heterogeneous parallelization over multi-core CPUs. Ipopt offers an interface to SSIDS.

Preparations have been going on over the last year for a new 2023 release of HSL. Significant work has gone into the software engineering aspects of testing, supporting and distributing HSL and facilitating its use. Beyond 2023, the directions HSL takes will reflect the evolving expertise and research interests of the Group, as well as the requirements of users and of projects within STFC that have need of computational mathematics.

Availability of HSL All usage of HSL packages (including by academics) currently requires a licence. Details of how to access the software are available at `https://www.hsl.rl.ac.uk/` (or contact `hsl@stfc.ac.uk`).

Acknowledgements I would like to thank Iain Duff, Nick Gould, Mike Hopper, Tyrone Rees, and John Reid for commenting on a draft of this report. Many individuals who are not named here have contributed to the Library at some time during the last 60 years. These include past and present Group members as well as external collaborators. Nevertheless, the maintenance and support for the Library has always relied on a very small core team, for whom it has been one element of their role at Harwell and/or RAL.

An overview of the career of the late HSL founder Professor Mike Powell may be found in [4].

REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Appl.*, 17(4):886–905, 1996.

[2] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23:15–41, 2001.

[3] J. Boyle, M. Mihajlovic, and J. Scott. `HSL_MI20`: an efficient AMG preconditioner for finite element problems in 3D. *Int. J. Numer. Method Engrg.*, 82:64–98, 2010.

[4] M. D. Buhmann, R. Fletcher, A. Iserles, and P. Toint. Michael J. D. Powell, 29 July 1936–19 April 2015. Technical Report NA2017/04, Cambridge University Numerical Analysis Group, 2017. `http://www.damtp.cam.ac.uk/user/na/NA_Group/Reports11_17.html` and `https://doi.org/10.1098/rsbm.2017.0023`.

[5] `https://www.numerical.rl.ac.uk/spral/`

[5] M. Campbell-Kelly. In praise of Wilkes, Wheeler, and Gill. *Communications of the ACM*, 54(9):25–27, 2011.

[6] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, volume 17. Springer Series in Computational Mathematics, 1992.

[7] A. R. Curtis and M. J. D. Powell. Program and subroutine library for STRETCH, 1963. Memo from Harwell Laboratory.

[8] A. R. Curtis and J. K. Reid. Fortran subroutines for the solution of sparse sets of linear equations. Technical Report AERE–R 6844, United Kingdom Atomic Energy Authority, 1971. Her Majesty's Stationery Office, London.

[9] T. A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Soft.*, 30(2):196–199, 2004.

[10] I. S. Duff. MA28–a set of Fortran subroutines for sparse unsymmetric linear equations. Technical Report AERE-R.8730, Harwell Laboratory, UK, 1980.

[11] I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Soft.*, 30(2):118–154, 2004.

[12] I. S. Duff and J. Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM J. Matrix Anal. Appl.*, 20(4):889–901, 1999.

[13] I. S. Duff and J. K. Reid. Some design features of a sparse matrix code. *ACM Trans. Math. Soft.*, 5(1):1–35, 1979.

[14] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Trans. Math. Soft.*, 9(3):302–325, 1983.

[15] I. S. Duff and J. A. Scott. The design of a new frontal code for solving sparse, unsymmetric systems. *ACM Trans. Math. Soft.*, 22(1):30–45, 1996.

[16] P. A. Fox, A. D. Hall, and N. L. Schryer. The PORT mathematical software library. *ACM Trans. Math. Soft.*, 4(2):104–126, 1978.

[17] N. I. M. Gould, D. Orban, and P. L. Toint. GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Trans. Math. Soft.*, 29(4):353–372, 2003.

[18] N. Hance. *Harwell – the Enigma Revealed.* Enhance Publishing, Oxfordshire, 2006.

[19] J. D. Hogg, J. K. Reid, and J. A. Scott. Design of a multicore sparse Cholesky factorization using DAGs. *SIAM J. Sci. Comput.*, 32(6):3627–3649, 2010.

[20] J. D. Hogg, J. K. Reid, and J. A. Scott. Guidelines for the development of HSL software, 2011 Version. Technical Report RAL-TR-2011-015, Rutherford Appleton Laboratory, 2011. `https://epubs.stfc.ac.uk/work/49289`.

[21] J. D. Hogg and J. A. Scott. HSL_MA97: a bit-compatible multifrontal code for sparse symmetric systems. Technical Report RAL-TR-2011-024, Rutherford Appleton Laboratory, 2011. `https://epubs.stfc.ac.uk/work/61445`.

[22] M. J. Hopper. Harwell Subroutine Library. A catalogue of subroutines. Technical Report AERE–R6912, Harwell, United Kingdom Atomic Energy Authority, 1971.

[23] M. J. Hopper. Harwell Subroutine Library. A catalogue of subroutines (1973). Technical Report AERE–R 7477, Harwell, United Kingdom Atomic Energy Authority, 1973. `https://archive.org/details/DTIC_ADA092033`.

[24] M. J. Hopper. Harwell Subroutine Library. A catalogue of subroutines (1973) Supplement No.1. Technical Report AERE–R 7477–SUPPL–1, Harwell, United Kingdom Atomic Energy Authority, 1974.

[25] M. J. Hopper. TSSD, a typsetting system for scientific documents. Technical Report AERE–R 8574, Harwell, United Kingdom Atomic Energy Authority, 1976.

[26] M. J. Hopper. Harwell Subroutine Library. A catalogue of subroutines (1973). Supplement No.2. Technical Report AERE–R 7477–SUPPL–2, Harwell, United Kingdom Atomic Energy Authority, 1977. `https://archive.org/details/DTIC_ADA092034`.

[27] M. Jewess. Harwell old and new: its renaissance as symbolised by the relocation of an RSC National Chemical Landmark plaque, 2019. Short essay in the Royal Society of Chemistry Historical Group Newsletter. No. 75.

[28] MUMPS. A parallel sparse direct solver. `http://mumps.enseeiht.fr/`.

[29] J. K. Reid and J. A. Scott. Guidelines for the development of HSL software, 2008 Version. Technical Report RAL-TR-2008-027, Rutherford Appleton Laboratory, 2008. `https://epubs.stfc.ac.uk/work/49084`.

[30] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[31] M. Wilkes, D. Wheeler, and S. Gill. *Preparation of Programs for an Electronic Digital Computer, Second Edition.* Addison-Wesley Press, 1957.