nag®

NAG and the NAG Library
The Code Contributors

Introduction

The NAG Library is a set of mathematical and statistical algorithms used by thousands around the world for the solution of numerical problems. Every single release of the Library has included numerical code contributed by professionals working in industry and academia. These esteemed *Code Contributors* generously give their code to help others gain benefit from their expert algorithms. Each code donated is then documented, tested, maintained and supported by NAG experts and engineered to run on different software and hardware configurations.

We acknowledge with gratitude the contributions of all these people to the development of the Library. This publication looks into the code contribution process and by interviewing people that have contributed code offers insight into the activity.

Interviewees

Fred Hickernell
Professor, Illinois Institute of Technology

Christopher Mower
MSc Student University of Manchester

Craig Lucas
NAG Senior Technical Consultant

Edvin Deadman
NAG Accelerator Software Engineer and former University of Manchester KTP Associate

Maurice Cox
National Physical Laboratory, UK

Rebecca Killick
Lecturer in Statistics, University of Lancaster

Klaus Schittkowski
Professor, University of Bayreuth

Mike Croucher
Research Software Engineer, University of Sheffield

Tell us a little about how the relationship with the Numerical Algorithms Group (NAG) and your team began, and how it has evolved to the present?

Craig Lucas

NAG Senior Technical Consultant – *speaking of his role in the NAG / University of Manchester KTP project*

Collaboration between the University of Manchester and NAG goes back to NAG's roots as a university software collaboration. The parties here go back to my PhD, as Nick Higham was my supervisor; although Nick worked with other NAG employees too. Knowledge Transfer Partnerships (KTPs) were suggested to the School of Maths and they are excellent ways to collaborate. As well as funding for the parties concerned they provide a framework to work in, with regular access to the academics. The glue in the KTP was the associate who would take their academic knowledge and embed it in industry; the NAG Library in our case. We employed Edvin Deadman who very successfully carried this role out, adding over 40 new routines to the Library. After the KTP Edvin worked for the university for 18 months and is now back at NAG, and will no doubt continue to work with Nick and colleagues in some way or another.

Edvin Deadman

NAG Accelerator Software Engineer and former University of Manchester KTP Associate

In October 2011, NAG and the University of Manchester began a three year 'Knowledge Transfer Partnership' (KTP). My role as the 'KTP Associate' was to implement the algorithms developed at the university by Professor Nick Higham's Numerical Linear Algebra Group into the NAG Library.

At the end of the project I continued to work with Nick Higham at the University of Manchester for 18 months as a post-doctoral research associate, before re-joining NAG in May 2015.

Craig Lucas

NAG Senior Technical Consultant – *speaking of his role in the Nearest Correlation Matrix team at NAG*

Development of Nearest Correlation Matrix (NCM) codes at NAG dates back to 2008 when our first routine was developed from an algorithm by Rüdiger Borsdorf during his MSc with Nick Higham. Rüdiger went on to do a PhD with Nick, funded by NAG, with myself as his industrial supervisor. During this time he wrote a new routine based on his research and I extended the original NCM routine.

Our customers still needed additional functionality and after talking to Defeng Sun who had written the original algorithm, with Qi that Rüdiger had improved I wrote a new one that included element-wise weights. I had some MATLAB® code to start with, which was an enormous help.

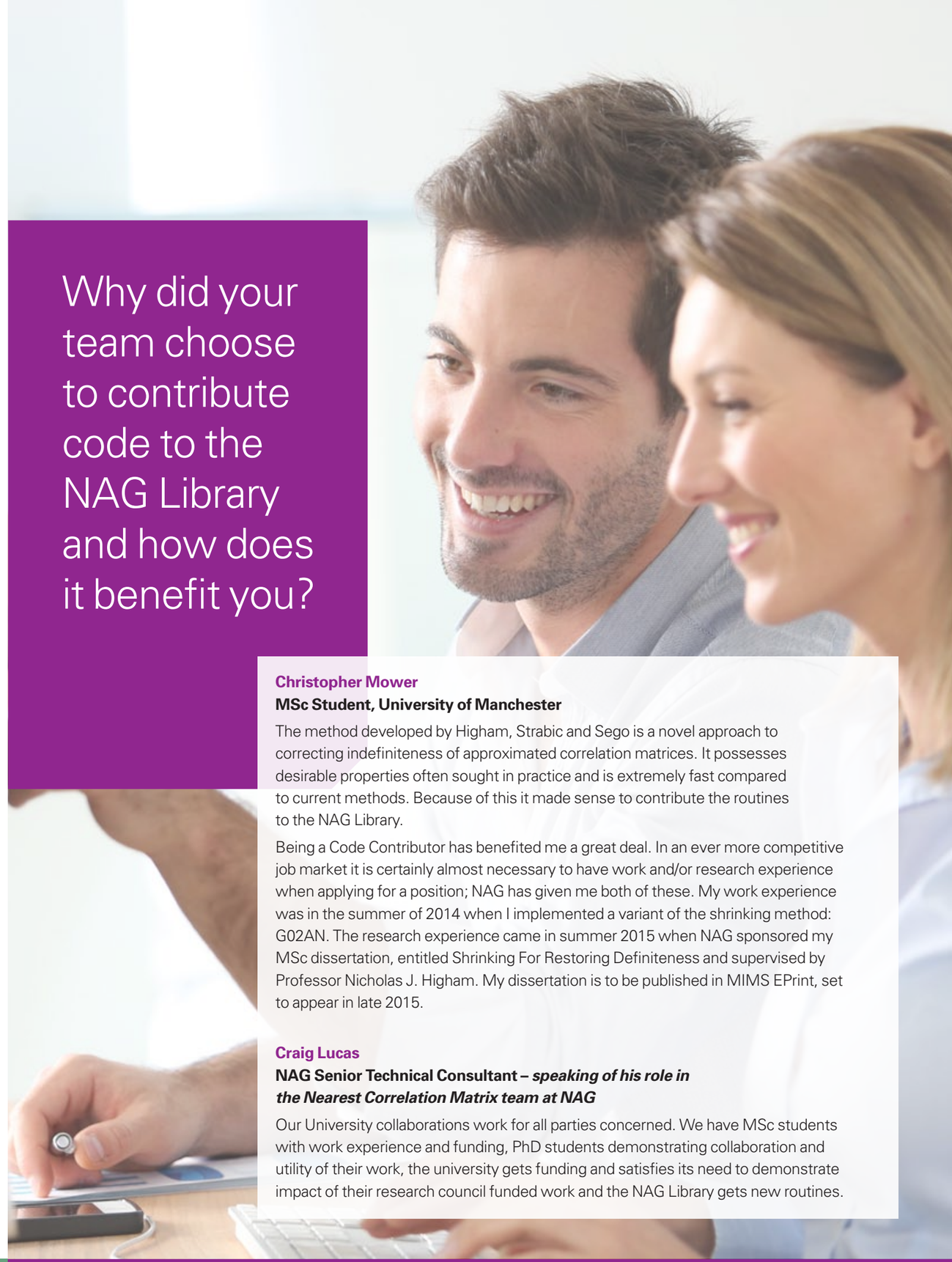
Two years later, a new student of Nick's, Natasa Strabic, started looking at a completely different shrinking algorithm for NCM. I had just met Chris Mower and with Natasa's help he wrote a new routine for the Library. I later extended this for Mark 26 to allow users to fix arbitrary elements.

Christopher Mower

MSc Student, University of Manchester

I first met Craig Lucas, Senior Technical Consultant at NAG while I was doing my MSc at the University of Manchester when he gave a talk to my class. Afterwards I had a chat with him and explained that I was looking for work experience and something interesting to do over the summer. About a month later I received an email saying that he had found a project that might interest me and could we meet for a chat. I met him at the University of Manchester, where we also met Natasa, who gave a presentation about a new Nearest Correlation Matrix method called shrinking to restore definiteness to approximated correlation matrices she, Professor Nicholas Higham and Vedran Sego had been working on.

Why did your team choose to contribute code to the NAG Library and how does it benefit you?



Christopher Mower

MSc Student, University of Manchester

The method developed by Higham, Strabic and Sego is a novel approach to correcting indefiniteness of approximated correlation matrices. It possesses desirable properties often sought in practice and is extremely fast compared to current methods. Because of this it made sense to contribute the routines to the NAG Library.

Being a Code Contributor has benefited me a great deal. In an ever more competitive job market it is certainly almost necessary to have work and/or research experience when applying for a position; NAG has given me both of these. My work experience was in the summer of 2014 when I implemented a variant of the shrinking method: G02AN. The research experience came in summer 2015 when NAG sponsored my MSc dissertation, entitled Shrinking For Restoring Definiteness and supervised by Professor Nicholas J. Higham. My dissertation is to be published in MIMS EPrint, set to appear in late 2015.

Craig Lucas

NAG Senior Technical Consultant – *speaking of his role in the Nearest Correlation Matrix team at NAG*

Our University collaborations work for all parties concerned. We have MSc students with work experience and funding, PhD students demonstrating collaboration and utility of their work, the university gets funding and satisfies its need to demonstrate impact of their research council funded work and the NAG Library gets new routines.

You contributed code as a team; can you describe how the development process worked within a group environment?

Edvin Deadman
NAG Accelerator Software Engineer and former University of Manchester KTP Associate

The process would begin with the Numerical Linear Algebra Group at the University of Manchester. The output of their research would typically be combinations of peer-reviewed papers containing new algorithms, and MATLAB code used to test them. From this research output, we would choose algorithms that we believed would be useful to NAG Library users. I would then implement them in Fortran, looking out for any potential improvements that could be made, and write the accompanying documentation.

Writing a new NAG routine is far from being the longest stage of the process. That honour goes to writing the test program for the routine! Each new routine has an associated test code which performs numerous checks to make sure the routine is robust, stable and gives the expected numerical results.

Craig Lucas
NAG Senior Technical Consultant – *speaking of his role in the NAG / University of Manchester KTP project*

The KTP gave the collaboration structure. The associate was required to understand existing research (as well as contribute to ongoing work) and write the code. The code would go through NAG's usual Peer Review process and the associate (Edvin) got support from the industrial supervisor, (me), and other NAG colleagues as you would expect.

Craig Lucas
NAG Senior Technical Consultant – *speaking of his role in the Nearest Correlation Matrix team at NAG*

Most of the routines mention here have had an academic paper and MATLAB code. A staff member at NAG then writes the Fortran, the testing code and documentation. When students have written work they do so under the supervision of someone at NAG.

Christopher Mower
MSc Student, University of Manchester

The relationship between NAG and the University of Manchester is beneficial for all involved. The Nearest Correlation Matrix project is a great example of how a number of parties can work together to achieve an end goal that benefits all in a unique and individual way. NAG adds to the increasing functionality of the NAG Library, NAG customers receive the benefit of the increased functionality and the university is able to clearly demonstrate the impact of its research conducted by its staff and students.

Could you talk a little about / describe the specific code/s that your team contributed to the Library?

Craig Lucas
NAG Senior Technical Consultant – *speaking of his role in the Nearest Correlation Matrix team at NAG*

All of the NCM routines do one thing. They take a matrix (a square and symmetric grid of numbers that are between –1 and 1, with 1's on the diagonal) and make it a true correlation matrix mathematically. We think of the input as an approximate correlation matrix, in that it looks like one but doesn't have the important mathematical property of being positive semidefinite (having eigenvalues that are greater or equal to zero). In practice some inputs may not be between –1 and 1, may not be symmetric and may not have 1's on the diagonal and the routines would fix that too.

The differences between them are mainly about fixing and weighting some of the original entries. Users want to apply their expert judgement. They know some entries are true and should remain unchanged. They may wish to emphasize some entries so they change less in the final output of the routines. They may also wish to strengthen the positiveness of the output, ensuring it is positive definite (that is, only has positive eigenvalue). This is important for further analysis.

So what we provide now is a suit of routines offering users plenty of choice.

Christopher Mower
MSc Student University of Manchester

The code restores positive semidefiniteness to an indefinite approximated correlation matrix through a transformation known as shrinking. The method takes the approximate, a target correlation matrix and finds a new correlation matrix that can be thought of as a compromise between the two extremes, which in some technical sense is optimal. The compromise is found by a modified bisection method and exploits properties of positive definite matrices to elicit fast execution times.

Edvin Deadman
NAG Accelerator Software Engineer and former University of Manchester KTP Associate

The routines I contributed concern matrix functions. You are probably familiar with 'normal' functions such as the exponential, sine, cosine or square root of a number. Well it turns out that it is possible to define, for example, the exponential of a matrix too. As it happens, this turns out to be very useful in all sorts of areas of science and engineering, from finance to theoretical physics. The Numerical Linear Algebra Group at the University of Manchester has developed many new algorithms for computing functions of matrices, and it is these algorithms that I implemented for the NAG Library. We now have a comprehensive suite of matrix function routines in Chapter F01.

Tell us a little about how your relationship with the Numerical Algorithms Group (NAG) began, and how it has evolved to the present?

Fred Hickernell
Professor,
Illinois Institute of Technology

Some years ago my student and I had written an algorithm to generate scrambled digital nets, which are used to improve the efficiency of Monte Carlo calculations. We came into contact with NAG, and they agreed to adopt the algorithm into the NAG Library.

Maurice Cox
National Physical Laboratory, UK

I was involved with the NAG Library since the early days and collaborated closely with Brian Ford, Jeremy du Croz, David Sayers, etc. In the 1980s I made a number of contributions to the NAG Library: interpolation, curve and surface fitting and linear algebra in particular.

Rebecca Killick
Lecturer in Statistics,
University of Lancaster

My relationship with NAG began when they invited me to give a seminar at their offices following my submission to the TakeAIM competition. During my visit they noted that they would like to include my work in a future release but that the current open source license I used prevented them from doing this. I subsequently gave them permission to use the code and have been encouraging people in my group to release their code to NAG ever since.

Klaus Schittkowski
Professor, University of Bayreuth

I've known about NAG for 30+ years. Moreover, I got in contact with one of the early representatives of NAG and some authors of the nonlinear optimization codes of NAG (P. Gill. W. Murray, M. Saunders) during my stay at Stanford University. My ongoing research in developing optimization algorithms and software and especially permanent cooperation with industrial companies from very different areas lead to various contacts with NAG users.

Why do you/did you choose to contribute code to the NAG Library and how does it benefit you?

Fred Hickernell
Professor, Illinois Institute of Technology

I think that it is important for good code to be widely available. At the same time, testing and maintaining code is a big job. NAG is well-situated to do both.

Maurice Cox
National Physical Laboratory, UK

I originally developed code for DASL, the NPL Data Approximation Subroutine Library, first of all in Algol and then FORTRAN. I decided to submit a number of items to the NAG Library, its stringent entry requirements benefited the software documentation and the test data sets immensely. That benefit ensued from the rigorous reviewing of my submissions by, mainly, Ian Gladwell. I used DASL a lot in my work and when subroutines were embedded in the NAG Library I used those as well or instead.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

I chose to contribute to NAG because I believe in making my work as widely available as possible. I distribute my code via R but this isn't used by everyone so the opportunity to widen take up of my work by donating code to NAG is wonderful. There is no direct quantifiable benefit to me but it may increase citation counts.

Klaus Schittkowski
Professor, University of Bayreuth

NAG is widely known especially in academia, but also in industry. Software becoming part of NAG might help to make the software and the author more visible.

Could you talk a little about/describe the specific code/s that you have contributed to the Library?

Fred Hickernell
Professor, Illinois Institute of Technology

Digital nets can be thought of as highly stratified samples of the unit cube. Earlier versions of this algorithm – developed by others – provided purely deterministic samples. We implemented a random scrambling scheme – introduced by others – that preserves the high degree of stratification. Our code in the NAG Library can be used to efficiently price financial derivatives, for example.

Maurice Cox
National Physical Laboratory, UK

I contributed subroutines for operating with polynomials in a Chebyshev basis: interpolation in 2D, least squares fitting in 2D and 3D, the former with constraints, differentiation and integration of a polynomial, and evaluation of standard uncertainties associated with a fitted polynomial and its derivatives and integrals. I also contributed subroutines for operating with splines in a B-spline basis, including the spline counterparts of the polynomial routines above. I think a *tour de force* was spline surface fitting for scattered data, when I put considerable effort into a special form of regularization to impose smoothness on the fitted surface and to make the code run as efficiently as possible. Today, the second effort would not be called for! I also contributed probably the first NAG Linear Algebra routines for dealing with structured systems, based on variable-bandwidth Cholesky decomposition.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

I have directly contributed the code for the PELT algorithm which detects changes in the underlying structure of a data set in a fast but exact way.

Klaus Schittkowski
Professor, University of Bayreuth

The code MISQP (Mixed-Integer Sequential Quadratic Programming) which entered the NAG Library at Mark 25, is based on a completely new mathematical approach to solve nonlinear mixed-integer problems. Its development has been sponsored by Shell over three years since they found out that for a large class of practical mixed-integer optimization problems, the existing algorithms are either not applicable (e.g. require relaxable integer variables, analytical model formulations, …) or are by far too expensive in terms of number of function evaluations, the main performance criterion in many engineering applications.

What do you see as the key benefit for the end user from accessing your methods via the NAG Library?

Christopher Mower
MSc Student, University of Manchester

In applications such as finance one may wish to compute an approximate correlation matrix which can be potentially indefinite and hence not a valid correlation matrix. Usually a nearest correlation matrix replaces the approximate and it is often desirable to weight or fix some elements known to be accurate, however this can be expensive to compute. The matrix computed via shrinking will be a valid correlation matrix but is not necessarily the nearest and shrinking is able to fix and weight elements. The main benefit users of the NAG Library will see is a vast difference in speed between current nearest correlation matrix algorithms and shrinking; the shrinking method, even for large matrices, is computationally inexpensive compared to existing nearest correlation matrix algorithms.

Fred Hickernell
Professor, Illinois Institute of Technology

The code is highly reliable. The Library contains some of the best algorithms available.

Maurice Cox
National Physical Laboratory, UK

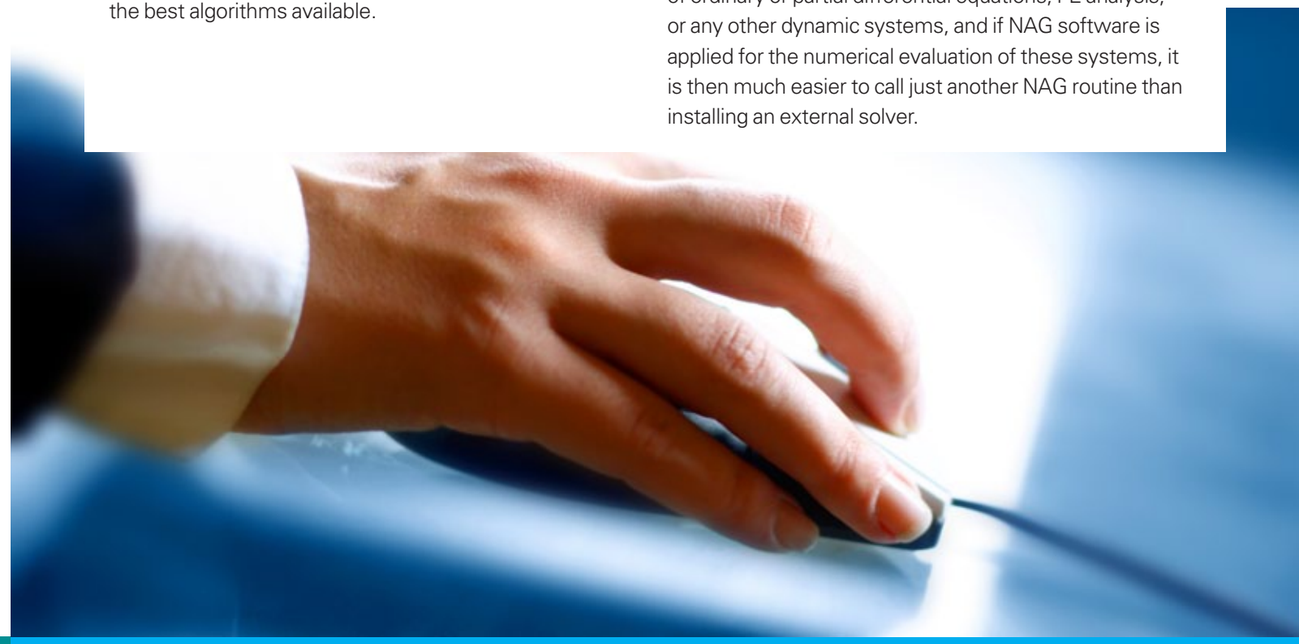
Key benefits are robustness of the Chebyshev and B-spline bases, and the ability to have polynomial degrees when necessary in the hundreds and an unlimited number of spline knots. Backward error-analyses exist for most of my contributed code: the solution provided is the exact solution of a closely neighbouring problem.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

Accessing my work via the NAG library gives it a 'seal of approval' and assures people that the code is accurate and correct.

Klaus Schittkowski
Professor, University of Bayreuth

Optimization requires the computation of objective function and constraint function values. If the underlying application is complex and requires, e.g., the solution of ordinary or partial differential equations, FE analysis, or any other dynamic systems, and if NAG software is applied for the numerical evaluation of these systems, it is then much easier to call just another NAG routine than installing an external solver.



How can NAG ensure customers and the community continue to benefit from code contributions such as yours?

Edvin Deadman
NAG Accelerator Software Engineer and former University of Manchester KTP Associate

The key is to continue our good relations and collaborations with academic institutions, as this is ultimately where most algorithms originate. It is clear how much NAG gets out of such collaborations. We need to ensure that academic institutions themselves continue to benefit from working with NAG.

Craig Lucas
NAG Senior Technical Consultant – *speaking of his role in the Nearest Correlation Matrix team at NAG*

Build relationships with research students through attending meetings, conferences, etc. Foster relationships with particular research groups. Make collaboration with NAG an obvious next step when people are developing algorithms.

Christopher Mower
MSc Student, University of Manchester

The best way NAG can continue to benefit its customers and communities, I feel, is to continue to sponsor MSc and PhD projects.

Maurice Cox
National Physical Laboratory, UK

At NPL, being a standards laboratory, traceability and provenance are key in our work. I consider that NAG's trademark is reliability, which should always be emphasized at every opportunity.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

NAG should have a campaign to directly ask researchers for their code. I never would have thought to donate code until I was asked.

Klaus Schittkowski
Professor, University of Bayreuth

Difficult. I mentioned before that many years ago, I got in contact with a NAG representative during a conference. Maybe, a more visible scientific representation of NAG could help, e.g., by motivating authors to cite NAG more significantly and by organizing more NAG sessions.

Do you think that producing good quality software is recognised as highly as academic research and papers are?

Edvin Deadman
NAG Accelerator Software Engineer and former University of Manchester KTP Associate

Producing good software is increasingly becoming part of academic research life. A huge number of academic papers, particularly in applied mathematics, physical sciences and engineering, go hand in hand with new pieces of software. Peers need access to this software in order to verify and expand on the results. Therefore it is now quite usual to see software, written as part of a research project, made available on, for example, GitHub. In addition, the usability and documentation for such software is improving.

That said, I do not think producing good software is recognised as highly as producing papers yet, but this is certainly changing. For example, several UK universities now have recognised 'Research Software Engineer' positions.

Fred Hickernell
Professor, Illinois Institute of Technology

It should be, but it may not be yet.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

No, even when you write a paper about your software and get lots of citations it isn't treated as highly as "proper" research.

Klaus Schittkowski
Professor, University of Bayreuth

Depends on the scientific tradition and visibility in the public. The answer is definitely yes in the US, e.g., and definitely no or maybe in Europe.

Do you think that students have the right skills to write quality software? And if they don't, how does this affect the community now and in the future?

Craig Lucas
NAG Senior Technical Consultant

No, in general. Students are driven by results. They write a code that will give them the right answer, in that it demonstrates a theory etc. They have probably thought very little about how others would use their code. How well will it behave on a similar problem for example? They are unlikely to have written test code or provided adequate documentation. They will be given no credit for writing the software for their research degree so this is understandable. They may be contributing to a department or community code. They may have quality control in place or they may not. The code will grow organically but not coherently. The student is unlikely to have training in code optimization and the tools to identify and fix inefficient aspects.

Fred Hickernell
Professor, Illinois Institute of Technology

They do not. This is something that we are trying to address at my institution, but it needs to be addressed by the whole computational science and engineering community. Computational mathematicians understand how to prove that algorithms work, but they do not know how to implement them. Computer scientists may know how to write quality software, but they do not understand mathematical software well.

Christopher Mower
MSc Student, University of Manchester

As an MSc student who has written quality software that has been implemented in the NAG Library I am evidence that students do have the right skills.

Rebecca Killick
Lecturer in Statistics, University of Lancaster

Undergraduate maths/stats students don't see the value in being able to code, they see this as a computer scientist's job. However, if they go on to get a job using their maths/stats skills they need to be able to code. Their code often doesn't have to be fast or perfect as they will be using the output to produce reports but they should be able to code for their own use. Sadly they don't see the value.

PhD students appear to see the value in being able to write good code and I personally encourage all my students to make their code available to the world. Hence it would be useful for them to be able to write quality software but the student needs to have the interest and ability for it to be useful to them.

Ultimately, if you are able to translate your work into a piece of software that other people can use, your work will live on. You can have the best theory and results in the world but unless people can use it it will be forgotten.

Klaus Schittkowski
Professor, University of Bayreuth

The answer to the first question is no, in general. It is extremely difficult to teach the required skills in special courses for many reasons. However, students have to write theses (Diploma, Master, Ph.D., …) and they learn from the supervisor or other available sources how to write qualified numerical software. But the supervisor has to possess the experience and the skills to write quality software, which is not available in general. However, I am quite sure that good students will easily pick up the required skills in a very short time if this turns out to be important.

A NAG user's point of view

Mike Croucher
Research Software Engineer, University of Sheffield

As a Research Software Engineer, I have worked with NAG for over a decade – first at the University of Manchester and, more recently at the University of Sheffield. My role requires me to be able to support a large number of software applications and programming languages used throughout numerous areas of scientific computing.

My style of software support centres around not being overly prescriptive with respect to the software and languages used by researchers – I will work in whatever system my collaborators find most comfortable while simultaneously attempting to 'nudge' them into adopting better practices. This approach has seen me working in systems and languages as diverse as MATLAB®, LabView, Mathematica, OriginPro, Python, R, Maple, Excel, C and Fortran on hardware ranging from low powered laptops up to some of the most powerful supercomputers in the country. I've worked with researchers in a huge range of fields including numerical algorithm development, medical image processing, radio astronomy, machine learning, finance, biostatistics, materials and land mine detection.

Across all of these fields of research and differing computational systems one thing has remained constant – I can rely on NAG to deliver results. The NAG Library is one of the most comprehensive collections of high quality, well documented, robust and efficient numerical routines available today. It can be used within every system and language that I support and is backed by one of the best support teams in the industry.

Time and time again, NAG have helped me make research code faster and more robust. Although superb, using the Library is just the beginning of it. NAG's true value shines through when you find yourself stuck in a numerical quagmire with no sign of a sane way out. Over the years NAG's team have helped support and educate me numerous times – they are the support behind the support!

My contributions to NAG's code base have been rather modest. I am most interested in ensuring interoperability between systems and, as such, focused on areas such as early experiments with using NAG from within Python. Now that NAG have a comprehensive and fully supported Python module, such experimentation is no longer necessary. I also give extensive feedback to NAG concerning how their routines are used in practice by researchers and how they could be improved. I find it very gratifying to see many of these suggestions implemented in new versions of the Library.

I feel that my relationship with NAG is a mutually beneficial partnership: a partnership that benefits NAG, me and, most importantly, the researchers I support.

nag®

NAG and the NAG Library
The Code Contributors

Numerical Algorithms Group

Wilkinson House
Jordan Hill
OXFORD OX2 8DR
UK

www.nag.com