12

October 1997

WGS Newsletter

Distributed by: **Working Group on Software WGS**

ISSN 0921-0490

Contents: **Page**

Contact address of the WGS: Mrs. Ida Tassens, Secretary of WGS
Katholieke Universiteit Leuven
Dept. of Electrical Engineering (ESAT)
Kardinaal Mercierlaan 94
3001 Leuven-Heverlee
Belgium

email: ida.tassens@esat.kuleuven.ac.be
phone: + 32 16 32 17 09
fax: + 32 16 32 19 70

1 Editorial

This is WGS newsletter 12. We are proud to announce you in this letter that SLICOT library release 3.0 is now available on our ftp or web site[1], and that our EC project NICONET has been approved for implementation during the next 4 years. Because NICONET can survive, all future WGS activities will be integrated in this project. Also for that reason, our next newsletter will be called NICONET newsletter. More information on that project, as well as a list of all partners involved, is given in Section 3.

The main activity of NICONET will be the further development of the SLICOT library and therefore we already implemented the routines, currently available in SLICOT release 2.0, on our WGS web and ftp site. From October 1996 on, Vasile Sima worked hard at the ESAT laboratory in order to upgrade, improve, standardize and implement all these routines electronically on our site. This labour-intensive task is now ready and we invite all of you to visit our web or ftp site and retrieve all routines you need in your current applications. Also, let us know your reactions (positive and negative) and suggestions for improvement or completion of the library. Detailed information on the present status of SLICOT and the most significant improvements is given in Section 2.

Since our new SLICOT routines are heavily based on LAPACK and its future release, we also like to give you more news on the status of LAPACK and its future release in Section 4.

Finally, we would like to thank Aloys Geurts and Ruth Kool from Eindhoven Technical University who stopped their WGS activities last summer. As you all know, Ruth has been our secretary during many years (since 1983), while Aloys (member of WGS since 1981) was the main person responsible for the inventory, standardization and documentation of the SLICOT routines. WGS looses two driving forces but thanks to NICONET their work will be continued by others: the WGS secretariat is now moved from Eindhoven to Leuven and Mrs. Ida Tassens will be our new secretary (see address on the cover page). Vasile Sima will be our main person responsible for the SLICOT library.

Sabine Van Huffel
Chairperson of WGS and Coordinator of NICONET.

[1] accessible through World Wide Web (URL http://www.win.tue.nl/wgs/) or via anonymous ftp (wgs.esat.kuleuven.ac.be/pub/WGS/SLICOT/).

2 SLICOT: now available on WWW

As announced in the Editorial of this Newsletter, SLICOT library release 3.0 is now freely available on the WGS ftp site, `wgs.esat.kuleuven.ac.be` (directory `pub/WGS/SLICOT/` and its subdirectories) in compressed (gzipped) tar files. On line `.html` documentation files are also provided there. It is possible to browse through the documentation on the WGS homepage at the World Wide Web URL `http://www.win.tue.nl/wgs/` after linking from there to the SLICOT web page and clicking on the `FTP site` link in the freeware SLICOT section. The SLICOT index is also operational. Each functional "module" can be copied to the user's current directory, by clicking on an appropriate location in the `.html` image. A "module" is a compressed (gzipped) tar file, which includes the following files: source code for the main routine and its test program, test data, execution results, the associated `.html` file, as well as the source code for the called SLICOT routines. There is also a file, called `slicot.tar.gz`, in the directory `/pub/WGS/SLICOT/`, which contains the entire library. The tree structure created after applying `gzip -d slicot.tar` and `tar xvf slicot.tar` is:

`./slicot/` — for routine source files;
`./slicot/doc/` — for html files;
`./slicot/tests/` — for test programs/data/results files.

2.1 Development of the public release of SLICOT

Converting the NAG SLICOT library 2.0 to a freely available software package, and adapting it to the new implementation and documentation standards, has been a time consuming activity, which also offered the opportunity to improve the codes as much as reasonably possible. This process, which resulted in *Release 3.0* of SLICOT, has evolved in several steps (not necessarily taken in sequence):

1. Removing the dependence of SLICOT routines on proprietary NAG routines, by replacing NAG routine calls by equivalent BLAS or LAPACK routine calls, and by implementing new mathematical routines, e.g., for solving linear systems with complex upper Hessenberg coefficient matrices, or computing QR- and LQ-decompositions of some structured or partitioned matrices.

2. Adapting the codes to the new LAPACK-like implementation and documentation standards. This was achieved by implementing new user interfaces (including CHARACTER-type option parameters) and new error handling schemes.

3. Improving the *modularity*, by restructuring the user-callable routines, functional overloading, etc.

4. Improving the *functionality*, by adding new features to increase the flexibility of usage.

5. Improving the *performance*, by reprogramming virtually all routines, turning the BLAS 1 and BLAS 2 calls into BLAS 3 calls whenever possible, using LAPACK block algorithms, exploiting any special problem structure, etc. The use of upper level BLAS routines resulted also in improved clarity and compactness of the source codes.

6. Improving the *reliability*, by replacing less stable calculations by mathematically equivalent, numerically stable ones.

The SLICOT Release 3.0 routines have been checked using adaptations of the previous test programs, data and results. Most part of the routines have also been tested by using the NAGWare Gateway Generator [2], which facilitates the integration of SLICOT into MATLAB. New MATLAB test programs have been written, which call the Fortran routines via these gateways.

2.2 Performance Results

This subsection summarizes performance results (efficiency, reliability, and accuracy), partially reported in [3], for some components of the new SLICOT release.

The results presented below have been obtained on a SUN Ultra 2 Creator 2200 workstation with 128 M RAM and operating system SunOS 5.5, by calling from MATLAB the gateways produced by the NAGWare Gateway Generator for the corresponding SLICOT codes. (The SLICOT routines have been compiled with `f77` using options `-O4 -native -u`.) These results show that SLICOT routines outperform MATLAB calculations. While the accuracy is practically the same, or better, the gain in efficiency by calling SLICOT routines can be significant for large problems. Note that the figures have been obtained by timing in MATLAB the equivalent computations. Even better efficiency is to be expected by calling the SLICOT Fortran routines only (not through the gateway), and similar accuracy/efficiency improvements have been obtained (or are possible) for other SLICOT computations, especially for large problems, due to the incorporated calls to upper level BLAS routines, and performant LAPACK blocked (and unblocked) algorithms.

The following tables indicate the accuracy (measured by either the relative errors or relative residuals, when available), and the time spent exclusively for equivalent SLICOT and MATLAB computations, for some routines in Chapters A, D, F, M and S of the library.

The **main conclusions** of the tests are:

1. The SLICOT gateways are usually several times faster than MATLAB.

2. The accuracy of SLICOT routines is at least as good as, or better than, that for MATLAB calculations.

3. Less memory is required by SLICOT routines for equivalent calculations, because the problem structure is fully exploited.

Typical results

(a) Table 1 displays comparative results for the SLICOT Fast Fourier transform routine `DG01ND` for real sequences, and the corresponding MATLAB function `fft`. For this purpose, random sequences `X` of length n were generated. Besides better efficiency, the accuracy of SLICOT routines shows an improvement which can be two orders of magnitude. The accuracy was measured by computing the distance between the original sequence and the inverse Fourier transform of the transformed sequence by

```
norm(X - ifft( fft( X ) ))/norm(X)
```

(which should theoretically be zero), and similarly for the SLICOT gateway. Note that `DG01ND` is 2–4 times faster than MATLAB.

Table 1: Comparison between DG01ND and MATLAB results.

n	Time		Relative error	
	DG01ND	MATLAB	DG01ND	MATLAB
1024	0.00	0.01	6.44e-16	6.56e-15
2048	0.00	0.01	1.06e-15	6.56e-15
4096	0.01	0.02	1.44e-15	2.00e-14
8192	0.02	0.05	1.95e-15	2.13e-14
16384	0.03	0.11	2.46e-15	8.21e-14
32768	0.07	0.27	2.81e-15	2.46e-13
65536	0.23	0.88	4.22e-15	4.26e-13

(b) Table 2 displays comparative results for the SLICOT linear systems solver MB02OD for triangular matrices, which includes a condition number estimator, and the MATLAB operation A\B. Triangular sets of linear systems, defined by $AX = B$, with $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$, were solved, where

```
A = triu(rand(n,n)) + eye(n)/2;  X = rand(n,m);  B = A*X;
```

Timing results, and relative error and residual norms of MB02OD and MATLAB results are given. Note that the relative errors are the "true" errors, because A and X have first been chosen and then used to compute $B = AX$, so the true results were known. For large matrices, the SLICOT gateway can be more than 5 times faster than MATLAB. The accuracy of MB02OD and MATLAB results is the same. The value 1 for the warning/error indicator info (an output parameter of MB02OD routine) shows that matrix A can be considered as numerically singular. Note that much poorer results have been obtained in MATLAB using the inv function for matrix inversion [3].

Table 2: Comparison between MB02OD and MATLAB results.

n	m	info	Time		Relative error	Relative residual
			MB02OD	MATLAB	MB02OD, MATLAB	MB02OD, MATLAB
64	16	0	0.01	0.02	4.75e-14	6.74e-17
128	32	0	0.03	0.12	1.39e-12	5.80e-17
256	64	0	0.17	0.99	1.97e-08	6.57e-17
512	128	1	1.69	9.42	1.14e-02	6.60e-17

(c) Tables 3 displays comparative results for the SLICOT Kalman filter routine FB01TD and MATLAB calculations. This routine updates the information square roots for upper Hessenberg matrices. The relative errors reported are the differences between SLICOT and MATLAB results. SLICOT code was about 4 times faster than MATLAB calculations, at comparable accuracy.

Table 3: Comparison between `FB01TD` and MATLAB results.

			Time		Relative error norms
n	m	p	`FB01TD`	MATLAB	`FB01TD`–MATLAB
16	4	8	0.00	0.01	`1.81e-16`
32	8	16	0.01	0.01	`3.10e-16`
64	16	32	0.03	0.11	`2.25e-16`
128	32	64	0.17	0.76	`1.86e-15`
256	64	128	1.48	6.48	`1.67e-15`

(d) Tables 4 and 5 display comparative results for the SLICOT subroutines `MB05MD` and `MB05OD`, for the computation of the matrix exponential, and the equivalent MATLAB functions `expm3` and `expm1`, respectively. While the algorithms implemented in SLICOT and MATLAB are basically the same, the implementation details differ significantly. The relative errors reported are the differences between the results obtained from SLICOT and MATLAB. SLICOT codes were usually faster (up to four times) than MATLAB calculations, at a comparable accuracy. Moreover, for upper triangular matrices, `MB05OD` can be much faster than `expm1`. An increase of over 30 times in speed has been achieved for $n = 256$.

Table 4: Comparison between `MB05MD` and MATLAB `expm3` results.

	Time		Relative error norms
n	`MB05MD`	MATLAB	`MB05MD`–MATLAB
16	0.01	0.01	`5.84e-15`
32	0.03	0.04	`1.08e-14`
64	0.18	0.33	`1.99e-14`
128	1.21	2.60	`1.82e-14`
256	9.29	34.04	`1.43e-13`

Table 5: Comparison between `MB05OD` and MATLAB `expm1` results.

	Time		Relative error norms
n	`MB05OD`	MATLAB	`MB05OD`–MATLAB
16	0.00	0.00	`1.57e-15`
32	0.01	0.02	`1.43e-14`
64	0.14	0.11	`3.61e-14`
128	1.28	0.80	`7.73e-14`
256	11.64	19.14	`1.66e-13`

(e) Table 6 displays comparative results for SLICOT routine `AB01ND`, for computing the controllability staircase form of a linear system in state-space form given by a matrix pair (A, B), $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and the MATLAB function `ctrbf` provided by the Control Toolbox [1]. Relative errors were computed using the following formula:

```
max(norm(z'*A*z - a)/norm(A),norm(z'*B - b)/norm(B))
```

where `a`, `b`, are the computed matrices in controllability staircase form, and `z` is the orthogonal transformation matrix. It should be mentioned that `ctrbf` uses singular value decompositions for rank determination while `AB01ND` employs rank-revealing QR factorizations. SLICOT routine has been up to ten times faster than `ctrbf` in the multi-input case ($m > 1$) and more than 250 times faster in the single-input case ($m = 1$).

Table 6: Comparison between AB01ND and MATLAB results.

		Time		Relative errors
n	m	AB01ND	MATLAB	AB01ND
16	2	0	0.01	4.75e-16
32	4	0.01	0.04	3.92e-16
64	8	0.04	0.22	6.05e-16
128	16	0.32	1.44	6.46e-16
256	32	2.56	25.10	1.35e-15

(f) Table 7 displays comparative results for the SLICOT pole placement routine `SB01MD` (for single-input systems), and MATLAB function `place`. Random systems and poles have been used. SLICOT routine has been much faster and more accurate. Frequently, `place` was not able to place the poles, resulting in an error message: "can't place eigenvalues there", while `SB01MD` still solved the problem well. The relative errors have been computed by the formulas (the first one not applicable for `place`)

```
max(norm(z'*(A - B*g')*z - a)/norm(A),norm(z'*B - b)/norm(B))
norm(sort(eig(a)) - sort(W))/norm(W)
```

where A, B, and a, b, define the original and transformed system, respectively, z is the transformation matrix, g is the computed gain vector, and W specifies the desired poles.

Table 7: Comparison between SB01MD and MATLAB results.

	Time		Rel. error	Rel. error eig.	
n	SB01MD	MATLAB	SB01MD	SB01MD	MATLAB
16	0.00	0.05	6.4392e-14	4.6955e-15	6.7700e-08
32	0.01	0.36	2.8780e-12	4.2186e-15	1.0983e-01
64	0.07	4.87	9.3902e-10	9.6121e-15	NaN
128	0.55	104.	4.3062e-07	1.3183e-14	NaN

(g) Table 8 displays comparative results for the SLICOT Lyapunov solver routine `SB03MD` (used here for continuous-time systems), and MATLAB function `lyap` (called for Lyapunov equations). Randomly generated matrices A and $C = C^T$ have beeen used. SLICOT routine has been 4–5 times faster than `lyap` and slightly more accurate.

Table 8: Comparison between `SB03MD` and MATLAB results.

	Time		Relative error	
n	`SB03MD`	MATLAB	`SB03MD`	MATLAB
16	0.01	0.04	1.3788e-14	1.7715e-14
32	0.03	0.17	1.0814e-13	1.4579e-13
64	0.21	1.13	2.3169e-13	7.8554e-13
128	1.39	8.35	5.3437e-13	8.4955e-13

(h) Finally, Table 9 displays comparative results for the SLICOT Sylvester solver routine `SB04MD` (used here for continuous-time case), and MATLAB function `lyap2`. The implemented algorithms differ (Hessenberg-Schur versus eigenvalue decomposition), but both are the fastest algorithms available in SLICOT and MATLAB for solving Sylvester equations. Randomly generated matrices A, B, and C have beeen used. SLICOT routine has been about four times faster than `lyap2` and more accurate.

Table 9: Comparison between `SB04MD` and MATLAB results.

		Time		Relative error	
n	m	`SB04MD`	MATLAB	`SB04MD`	MATLAB
16	8	0.00	0.02	2.8896e-14	8.4219e-14
32	16	0.02	0.05	2.0149e-14	4.1296e-14
64	32	0.11	0.39	3.3743e-13	6.7799e-13
128	64	0.88	3.85	4.8597e-13	7.3803e-13

References

[1] *Control System Toolbox User's Guide.* The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, MA 01760, 1996.

[2] *NAGWare Gateway Generator, Release 2.0.* The Numerical Algorithms Group, Wilkinson House, Jordan Hill Road, Oxford OX2 8DR, UK, 1994.

[3] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT — A Subroutine Library in Systems and Control Theory. To appear in *Applied and Computational Control, Signals, and Circuits*, 1997. Also available as NICONET Report 97-03 from `wgs.esat.kuleuven.ac.be`, directory `pub/WGS/REPORTS/nic97-3.ps.Z`.

Vasile Sima, Research Institute for Informatics, Bucharest, Romania

3 NICONET: ready to start again

3.1 Introduction and Objectives

The future developments of SLICOT will be integrated in a thematic "Numerics in Control" network, entitled *Network for development and evaluation of numerically reliable software in control engineering and its implementation in production technologies.* This thematic network is part of the specific European Community programme on Industrial and Applied technologies (BRITE-EURAM III) under the fourth framework programme (1994-1998) on "Research and technological development" (RTD) and aims to promote scientific and technological co-operation and integration between European industry, research bodies and universities by linking on-going European non-Community funded RTD activities. This type of network is developed in two steps: a recommended exploratory phase of six months followed by the implementation phase.

The Commission of the European Union has given an exploratory award to WGS from October 1996 till April 1997 in order to start up this Numerics in Control Network – with acronym NICONET[2] – in the field of Industrial and Materials Technologies. The objectives of NICONET can be summarized as follows:

- to *intensify* the research in and collaboration on Numerics in Control in which European teams play a prominent role on a world scale. This is achieved by stimulating the collaboration of control specialists and numerical scientists, by exchanging specialists between academic and industrial research units and by the organisation of information exchange between academic and industrial research units within Europe.
- to *integrate* the SLICOT and RASP control libraries into a joint library, to *extend, improve* and *benchmark* it and to *adapt* it for easy implementation in general purpose CACSD packages.
- to *ensure the transfer* of information technology related to control of industrial processes to industry. To *facilitate access* to high technology software and convince industrial developers of the feasibility of this software, and the benefits in using it.

During this phase, WGS has attracted new partners with complementary expertise as well as industrial companies in order to prepare the implementation phase. At present, NICONET consists of the following 17 European partners.

Partners of the exploratory phase of NICONET:

1. **KUL-SISTA** Katholieke Universiteit Leuven, Department of Electrical Engineering, Research Group SISTA, Leuven, Belgium.
   Contact : Prof. Sabine Van Huffel
2. **UCL-CESAME** Université Catholique de Louvain, Centre for Systems Engineering and Applied Mechanics, Louvain-la-Neuve, Belgium.
   Contact : Prof. Paul Van Dooren

[2]For more information, see the NICONET world wide web homepage `http://www.win.tue.nl/wgs/niconet.html`

3. **TU Eindhoven** Eindhoven University of Technology, Eindhoven, The Netherlands :
   (a) Department of Mathematics and Computing Science.
   Contact : Dr. Anton Stoorvogel
   (b) Department of Electrical Engineering.
   Contact : Dr. Ad van den Boom

4. **TU Delft** Delft University of Technology, Delft, The Netherlands, Department of Electrical Engineering, Systems and Control Engineering Group.
   Contact : Dr. Michel Verhaegen

5. **DLR** Deutsches Zentrum für Luft- und Raumfahrt e.V. Oberpfaffenhofen, Germany.
   Contact : Dr. Andras Varga

6. **NAG** The Numerical Algorithms Group Ltd, Oxford, United Kingdom.
   Contact : Mr. Sven Hammarling

7. **TU CZ** Technical University Chemnitz-Zwickau, Numerical Analysis Group, Chemnitz, Germany.
   Contact : Prof. Volker Mehrmann

Additional Partners of the implementation phase of NICONET:

8. **Universidad Politecnica de Valencia** , Departamento de Sistemas Informaticos Y Computacion, Valencia, Spain.
   Contact : Prof. Vicente Hernandez

9. **INRIA** , Department META2, Scilab group, le Chesnay Cedex, France.
   Contact : Dr. François Delebecque and Dr. Serge Steer

10. **Universität Bremen** , Fachbereich Mathematik und Informatik, Bremen, Germany.
    Contact : Dr. Peter Benner

11. **University of Leicester** , Department of Engineering, Control System Research Group, Leicester, United Kingdom.
    Contact : Dr. Dawei Gu and S.K. Spirgeen

12. **Umeå University** , Department of Computing Science, Umeå, Sweden.
    Contact : Prof. Bo Kågström

13. **TBZ-Pariv GmbH** , Chemnitz, Germany.
    Contact : Dr. Riener Wohlgemuth

14. **LMS International** , Leuven, Belgium.
    Contact : Dr. Herman Van Der Auweraer

15. **Aspen tech** Eindhoven office, Best, The Netherlands.
    Contact: Dr. Jobert Ludlage and Prof. Ton Backx

16. **SFIM Industries** , Massy Cedex, France.
    Contact: Ir. Pierre Coustal

17. **Omron Spain** , Madrid, Spain
    Contact: Ir. Fernando Colás

In April 1997 these partners submitted a new proposal for the final implementation of the network with a detailed workplan over the next 4 years. This proposal has recently been approved by the European Communities. Negociations with the EC are still going on before the contract can be signed. It is expected that the extended network can start in the beginning of next year. The future development of SLICOT is thus guaranteed by the approval of this European "Numerics in Control" network NICONET. Its aim is to improve and complete SLICOT in such a way that it can be applied to a wide range of industrial applications of control and systems theory. This will be accompanied by the development of special routines for large-scale problems frequently encountered in practice and by embedding SLICOT in user-friendly environments such as MATLAB, Scilab (developed by NICONET partner INRIA, public domain and mainly used in Europe) and ANDECS (developed by NICONET partner DLR, built on the RASP subroutine library and mainly used in Germany). More details are given below.

3.2 Objectives and exploratory phase of NICONET

It is clear that the efforts to develop SLICOT are very intensive. Therefore, as mentioned in WGS newsletter 9, a coordinated future development of RASP and SLICOT into a joint library has been established to reduce the implementation efforts. In order to extend these RASP/SLICOT coordination efforts to other European software development initiatives in the area of numerical control, the thematic network NICONET has been set up recently.

The results of a questionnaire performed during the NICONET exploratory phase confirm the present need for performant numerical software and greatly support coordination of this activity on a European level. These results reflect the opinion of 55 universities or research centers and 17 industrial companies spread over Europe and can be summarized as follows (for more details, see WGS newsletter 11). The expansion of the present WGS network (NICONET partners in the exploratory phase) to a European level is strongly encouraged in order to obtain a wider base of software developers and potential users. However, quality control on the software made available must be imposed. Therefore, feedback from others than the developers is absolutely necessary. The network will focus on the development of numerically reliable and efficient control related software freely available and embedded in a user-friendly environment such as MATLAB in order to guarantee its widespread use in both academia and industry. A homogeneous user-interface is of high importance: most software users want to have flexible and powerful, but easy-to-use tools, and are willing to sacrifice speed for ease of use. However, people do want reliable answers, and therefore reliable software is needed. In addition, the library should be benchmarked and validated by means of real industrial examples. Commercial support should be provided too, especially for industry. The design of a library for large-scale applications will complement SLICOT. In this context high-performance computer tools are appropriate (software packages like BLAS, LAPACK, and new parallel packages should be used). In addition, the use of electronic means is widely accepted as the most flexible and user-friendly way to enhance information exchange and cooperation within the network. Finally, it should be noted that all but one replier of the questionnaire use MATLAB which is mainly due to the user-friendliness of the product. Also, the MATLAB toolboxes are very popular (used by more than 90% of the repliers). More than 30 software contributions have been reported in diverse control areas, in particular in system identification, optimal control, model reduction, time and frequency response. Despite these contributions, almost all repliers expressed their needs for new software (preferably in MATLAB or Fortran 77) in all areas of systems and control and pointed out the lack of tools to increase the efficiency of software development in process control applications. To realize these aims, NICONET will focus on the activities described in the following sections.

3.3 Development of performant numerical software for CACSD

In the past, WGS essentially relied on the expertise and software developed by its own members. Each contributor had its own focus, so that the present library only covers part of the whole field. Therefore the list of NICONET participants has been expanded to a representative European network and new research centers and universities with complementary expertise in software development in numerics of control have been selected which jointly cover almost the whole discipline of systems and control theory. In this way, the resulting joint library is potentially able to approach a "mature" status with respect to *size, completeness, and quality.* The present SLICOT subroutine collection is too restricted for broad industrial use and therefore NICONET will fill these gaps by starting to complete the library in industrially relevant areas for which the partners of NICONET have readily available prototype software and/or algorithms. These are basic numerical tools for control, model reduction, subspace identification, robust control, and nonlinear systems.

The future development of SLICOT will be accompanied by the development of a parallel version of SLICOT (with working title *PSLICOT*) for distributed memory computing environments. Parallelisation clearly opens new perspectives for solving large scale problems which frequently arise in industrial practice. As not every control problem requires the computing power of such machines, only those subroutines will be contained that can be used to solve control problems capable of taking advantage of distributed memory parallel computing environments. The set up of this parallel SLICOT version involves the extension of the SLICOT standards for high-performance computations on parallel architectures and the selection of standard communication kernels such as MPI or PVM and parallel numerical linear algebra libraries such as ScaLAPACK or PLAPACK.

3.4 Integration of software in a user-friendly environment

The main aim of WGS is to see the library be used by as many scientists and engineers in industry as possible, so that the careful efforts of the contributors bear fruit. This requires a wider distribution, and in order to guarantee this, a better integration of SLICOT in a user friendly environment is needed. If the software is not easy to use it will not be used, certainly not in industry! Those environments are chosen which are most commonly used by European industrial companies and research centers, namely MATLAB, Scilab, and ANDECS. Scilab has the advantage to be public domain. Very often industrial enterprises are not willing to pay additional library licences. These may be very expensive (especially for industry). For those companies Scilab can be a useful tool. A first step in this direction is the use of a compiler that automatically passes the function parameters from the CACSD environment, such as MATLAB, to any Fortran routine of SLICOT and back (e.g., the NAGWare Gateway Generator described in WGS newsletter 9), which clearly makes the routines of a Fortran library available to a broader group of users. Therefore, NICONET plans the development of MATLAB toolboxes based on calls of SLICOT routines by means of this tool.

3.5 Benchmarking and testing the software in an industrial environment

There is a definite need for more and adequate benchmarks for control methods and their implementation. These benchmarks should be practically oriented. Carefully chosen benchmarks give insight in the state of the art with respect to the performance of methods in the language of the control system analyst or control engineer. The need for this kind of insight is rapidly increasing due to today's widespread availability of a wealth of methods and implementations. Therefore, a SLICOT benchmark library and accompanying standards will be set up and made available through the WGS ftp site. To assess the performance, reliability, and versatility of SLICOT in industry, industrial enterprises specialised in the development of industrial software for implementation in production technology have been selected which jointly cover a broad spectrum of industrial applications, such as aerospace and automotive technology, robotics, and manufacturing. By integrating the new tools, provided through NICONET, into their products, these partners would be able to improve traditional production processes and make the much needed software available to industry. In addition, this facilitates largely future implementations of advanced solutions into production systems.

3.6 Information dissemination and access to control software

In order to facilitate dissemination of the SLICOT software and its transfer to a wide range of users, WGS, DLR and NAG decided to make SLICOT freely available. The use of electronic means is most appropriate to ensure easy and worldwide access to the library and therefore the software, together with the documentation, has been made available on the WGS ftp site (see Section 2 for details). It will also be available on CD-Rom or tape. Because of using the highly standardized Fortran 77 as programming language, SLICOT can be used under all major operating systems and platforms. Also, its manual should be available in paper form. These services, as well as commercial support, will be provided by NAG. As pointed out in the results of the NICONET questionnaire, this is of major importance for a lot of industrial users who want better services for maintenance of the product, advice for troubleshooting and software guidance.

Information exchange will be realized by the issue of an electronic NICONET newsletter. These issues, containing information on NICONET, SLICOT, and other CACSD software, are made available on the ftp and web sites and will be announced via appropriate electronic newsletters (such as the NA-NET News Digest[3] and the E-letter on Systems, Control, and Signal Processing[4]). The establishment of these electronic services, together with the set up and maintenance of electronic mail reflectors, the NICONET world wide web pages, the electronic repository of NICONET related reports, are major assets for the partners of NICONET and provide ready information exchange, up-to-date sources of information and software, and a means to publish and make available important software and results.

In addition, workshops around the topic of control algorithms and software, as well as tutorials and training courses, will be organized.

Sabine Van Huffel, ESAT-SISTA, Dept. EE, Katholieke Universiteit Leuven, Belgium

[3] For information about NA-NET (editor: C. Moler): mail to "na.help@na-net.ornl.gov".

[4] For information about this E-letter (editors: A. Stoorvogel and S. Weiland): send an (empty) e-mail message to "eletter@win.tue.nl" carrying the subject "info".

4 LAPACK and Related Developments

4.1 Introduction

Following the plans to make SLICOT into a freeware library, the WGS has been actively ensuring that the linear algebra in the library is based, as far as possible, upon the LAPACK software. In this short article, I give an introduction to LAPACK, indicate the contents of the next release, and also mention the BLAS Technical Forum, which is working to update the Basic Linear Algebra Subprograms (BLAS) in the light of modern languages, software and hardware.

4.2 LAPACK

LAPACK stands for Linear Algebra Package, and is a freely available package for the solution of dense and banded linear algebra problems aimed at PCs, workstations, and shared memory high-performance computers. The software can be downloaded from netlib (at the World Wide Web URL http://www.netlib.org/lapack/index.html) and is fully described in the LAPACK Users' Guide [1].

In order to make the package both efficient and portable, extensive use has been made of the Basic Linear Algebra Subprograms (BLAS), [6, 4, 3]. Many of the algorithms utilised by LAPACK were restructured into block-partitioned algorithms so that the Level 3 BLAS [3], aimed at matrix-matrix operations, could be used wherever possible for the computationally intense parts of the algorithms. Where this could not be achieved, the Level 2 BLAS [4], aimed at matrix-vector operations, have been used wherever possible.

By using LAPACK, users of SLICOT benefit from state of the art, numerically stable, efficient and reliable software for linear algebra, thus increasing the robustness and the efficiency of the control methods.

4.3 LAPACK, Release 3

It is intended to have a further release of LAPACK, Release 3, around the end of 1997. This next release is expected to include improvements, or new coverage in the areas of:

- a divide-and-conquer routine for the singular value decomposition, together with an associated least squares solver
- new simple and expert drivers for the generalized nonsymmetric eigenproblem
- new drivers for the generalized symmetric eigenproblem
- a faster QR decomposition with column pivoting
- a faster solver for the rank-deficient least squares problem
- improved linear system expert drivers
- a faster routine for the reduction of a banded matrix to tridiagonal form,

as well as number of other improvements to existing routines.

4.4 ScaLAPACK

As mentioned above, LAPACK is aimed at shared memory machines. ScaLAPACK, which stands for Scalable Linear Algebra Package, is the result of a project concerned with porting LAPACK to distributed memory parallel computers, as well as networks of PCs and workstations. ScaLAPACK uses the message passing paradigm, and in particular the software can be run on any machine for which MPI ([7]) or PVM ([5]) are available, and has been designed to look as similar to LAPACK as possible, and to utilise the LAPACK algorithms wherever that is sensible. ScaLAPACK currently contains a subset of LAPACK, but development is ongoing. As with LAPACK, the software is freely available and can be downloaded from netlib (http://www.netlib.org/scalapack/index.html) and is fully described in the ScaLAPACK Users' Guide [2], which is accompanied by a CD containing the software and several prebuilt library versions of ScaLAPACK.

Should it be decided to port SLICOT to distributed memory machines, then it should be straightforward to replace calls to LAPACK with calls to ScaLAPACK.

4.5 The BLAS Technical Forum

Following a workshop held in Knoxville in November 1995, a Technical Forum has been established to consider expanding the BLAS in a number of directions in the light of modern software, language, and hardware developments. The first meeting of the Forum was held in Nashville on February 19-20, 1996, and several meetings have taken place since.

Working groups have been established to consider the overall functionality, a lightweight interface, an object based interface, as well as issues such as parallel processing, sparse operations, data structures, extensions to the existing BLAS and language binding issues.

Information on the activity is available from the BLAST Forum home page at:

http://www.netlib.org/utk/papers/blast-forum.html

as well as information from the Knoxville workshop, and information on email discussion lists for those interested in participating, or monitoring the discussion. Input to the Forum is most welcome.

References

[1] E. Anderson, Z. Bai, C. H. Bischof, J. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen. *LAPACK Users' Guide.* SIAM, Philadelphia, PA, USA, 2nd edition, 1995. (Also available in Japanese, published by Maruzen, Tokyo, translated by Dr Oguni. URL: http://www.netlib.org/lapack/lug/lapack_lug.html).

[2] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. W. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide.* SIAM, Philadelphia, PA, USA, 1997. (Includes a CD containing the software, an HTML version of the Guide and LAPACK Working Notes. URL: http://www.netlib.org/scalapack/slug/scalapack_slug.html).

[3] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 16:1–28, 1990. (Algorithm 679).

[4] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson. An extended set of FORTRAN Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 14:1–32, 399, 1988. (Algorithm 656).

[5] A. Geist, A. Beguelin, J. J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing.* MIT Press, Cambridge, MA, USA, 1994.

[6] C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for FORTRAN usage. *ACM Trans. Math. Software*, 5:308–323, 1979.

[7] M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. J. Dongarra. *MPI: The Complete Reference.* MIT Press, Cambridge, MA, USA, 1996.

Sven Hammarling, NAG Ltd, Oxford, United Kingdom