9

ISSN 1373-6787

July 2002

NICONET Newsletter

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

Contents: **Page**

Contact address of the WGS: Mrs. Ida Tassens, Secretary of WGS
Katholieke Universiteit Leuven
Dept. of Electrical Engineering (ESAT-SCD(SISTA))
Kasteelpark Arenberg 10
3001 Leuven-Heverlee, Belgium
email: ida.tassens@esat.kuleuven.ac.be
phone: + 32 16 32 17 09 and fax: + 32 16 32 19 70

1 Editorial

Welcome to the nineth issue of the NICONET newsletter which informs you about the evolution of the SLICOT library and its integration in user-friendly environments such as `Scilab` and MATLAB, as well as about other NICONET activities related to CACSD software developments.
Since July 1, 2002, our EC thematic network project came to its end and at present we are finalizing our EC reports. We hope that our activity can be funded in the 6th framework. Therefore, we submitted an Expression of Interest in June in order to improve our chances for setting up or joining a network of excellence. Calls are coming out in the fall of this year. In the meantime maintenance and further development of the SLICOT library will be guaranteed by our international society, also called NICONET, which is operational since September 2001. Any funding received through this society will be used for the further development of the SLICOT library, as well as for the promotion and dissemination of the SLICOT software.
Sections 2 to 6 present as usual the new updates of the SLICOT library in subfields of systems and control. In Section 7, the use of the SLICOT model reduction tools in solving large-scale industrial modelling problems, is described. Section 8 includes a Call for papers for a Special Issue of the journal *Linear Algebra and Its Applications* on *Order Reduction of Large-Scale Systems*, which is a main topic of the NICONET thematic network project; moreover, two members of the NICONET team are special editors for this Special Issue. Finally, Section 9 gives more details about the newest additions to the SLICOT library, new reports and forthcoming events.

I hope you enjoy reading this newsletter.

Sabine Van Huffel
NICONET coordinator

2 Basic numerical SLICOT tools for control

2.1 Standard and generalized state space systems and transfer matrix factorizations

Several SLICOT routines performing computations for standard and generalized state space systems and transfer matrix factorizations, including, e.g., `AB13FD` and `SB02OD`, have been updated, and new routines have been developed and made available on the SLICOT ftp site.

The newly added routines include a solver for either the continuous-time or discrete-time algebraic Riccati equations for descriptor systems (`SG02AD`), as well as several routines for computing the transfer function matrix of a state-space representation (A, B, C, D) of a linear time-invariant multivariable system, using the pole-zeros method. Each element of the transfer function matrix is returned in a cancelled, minimal form, either as a polynomial ratio (with numerator and denominator polynomials stored either in increasing or decreasing order of the powers of the indeterminate), or in a pole-zero-gain form. The corresponding user-callable routines are `TB04BD` and `TB04CD`, respectively. Several lower-level routines have also been written and posted on the SLICOT ftp site.

In addition, two new MEX-files and several associated M-files are available for (partial) pole assignment, and for the computation of the periodic Hessenberg or periodic Schur decomposition of a matrix product, using orthogonal transformations. Also, high-level interfaces for Riccati benchmark collections have been made available in February 2002.

More details about these new developments are given in the file `Release.Notes`, stored on the SLICOT ftp site,

`ftp://wgs.esat.kuleuven.ac.be`

directory `pub/WGS/SLICOT/`.

Vasile Sima, Andras Varga and Paul Van Dooren

2.2 Structured matrix computations

The SLICOT routines for structured matrix computations have been finalized in 2001, and they were described in the previous issues of this Newsletter, as well as in the SLICOT Working Note 2000-2, revised in June 2001. This task continued with the further development of user-friendly MATLAB/Scilab interfaces for the newly developed routines, in particular, for QR factorization of block Toeplitz matrices. These interfaces will be made available on the SLICOT ftp site at the next library update.

A new report concerned with benchmarks for model reduction (SLICOT Working Note 2002-2: Y. Chahlaoui and Paul Van Dooren: A collection of benchmark examples for model reduction of linear time invariant dynamical systems) contains models with the polynomial structure obtained for second order mechanical systems.

Daniel Kressner, Vasile Sima and Paul Van Dooren

3 SLICOT tools for model reduction

3.1 SLICOT tools for controller reduction

The SLICOT routines and the associated toolbox for controller reduction have been finalized in 2001, and they were described in the previous issues of this Newsletter, and in several SLICOT Working Notes. Recently, the WEB page for this toolbox has been updated.

Andras Varga

3.2 SLICOT tools for model reduction of high order systems

3.2.1 Direct methods for model reduction

The work on direct methods has been completed, and the corresponding toolbox will be soon available at the NICONET website. Summarizing, over thirty new parallel routines have been implemented, standardized and documented. The main routines are presented in the next table (which includes user-callable and auxiliary routines solving general matrix problems).

Routine	Purpose
PAB09AD	Computes reduced (or minimal) order balanced models using either the **SR** or the **BFSR** B&T method
PAB09BD	Computes reduced order models using the **BFSR** SPA method
PAB09DDS	Computes a reduced order model by using the singular perturbation formulas
PMB03UD	Computes the SVD of a triangular matrix.
PSB03OT	Solves a Standard Lyapunov Matrix Equation. It covers continuous- and discrete-time systems and also transpose and transpose-free versions.
PDGEES	Reduces a matrix to Schur form. ScaLAPACK does not provide a complete driver routine for the reduction to the Schur form of a general matrix. A routine has been implemented for this operation, including the routines required for the computation of the transformation matrix from the reflectors.

During the last 6 months, the work has then been oriented to end with the parallelisation of the main SLICOT routines for reduction of stable systems (mainly `AB09AD, AB09BD, AB09DD`).

For this work it has been necessary to parallelise several auxiliary routines, namely `MB03UD` (routine in charge of computing the Singular Value Decomposition), `TB01ID` (reduce the 1-norm of the system matrix by using a diagonal similarity transformation applied iteratively), and `TB01WD` (reduce the system state matrix to upper real Schur form, applying the transformation to the other matrices), among others. Some of these routines were started in the last year, but most of them have been finished in this last period of 6 months.

Once these auxiliary routines have been parallelised, the parallelisation of the main routines has been possible. First, routines which work with the system in real Schur form and with the 1-norm reduced have been parallelised (`AB09AX, AB09BX` which have become `PAB09AY, PAB09BY`). And finally, the main user routines have been parallelised, which call all other routines.

As an example of the performance of these new routines, figure 1 shows the execution time of the PAB09AY routine (parallel version of AB09AX SLICOT routine). The experiments were executed on a 5-biprocessor 866 MHz Pentium III PC system linked with a Fast Ethernet and Gigabit Ethernet LAN (each biprocessor was used as a single processor). This routine, instead of the user-callable PAB09AD, has been chosen to show performance, because the time in it is more depending on newly developed routines. For the PAB09AD routine, the execution time is influenced by the ScaLAPACK routine in charge of transforming a Hessenberg matrix to real Schur form (PDLAHQR), thus hiding the times of the newly developed routines.

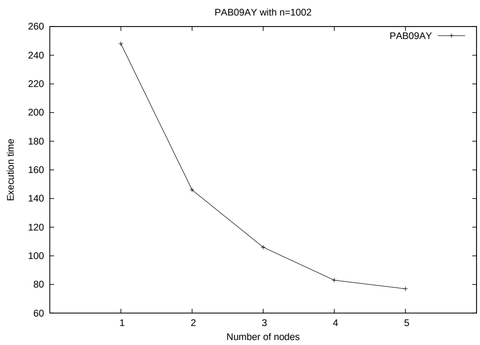

Figure 1: Time spent reducing a system in Schur form in parallel for different configurations.

This figure is an example of what is possible when using parallel algorithms. The execution times and memory requirements are reduced with respect to the sequential implementations, thus allowing to work with high order systems.

Vicente Hernandez, David Guerrero

3.2.2 Model reduction based on iterative solvers for computing the system Gramians

All work has been finalized at the end of the 2001 year. Additional benchmark testing has been done, but the results will be reported later.

Peter Benner

4 SLICOT tools for subspace identification

4.1 Standard software for nonlinear state space model identification

The previous phase of this task concentrated on the identification of multivariable nonlinear Wiener systems. Such systems are a concatenation of a linear dynamic block followed by a static nonlinearity. An inventory of existing algorithms focused on schemes that enable the linear part to be represented in state space form. This inventory lead to the selection of a combination of a variant of the subspace identification method and a single layer neural network to model the static nonlinearity.

The standardization of the associated, planned routines for identification of nonlinear, state space systems has been completed in 2001, but further polishing appeared as necessary for improving the reliability and speed. The planned improvements have been performed, and further work has been done in parallel on the other subtasks of the Task III.B (integration of the routines in MATLAB and Scilab, selection of benchmark problems, extension of the toolbox).

A list of the routines and a brief description of their functionality is given below. The list includes (part of) the related mathematical and transformation routines which have been developed for the Task III.B. Most part of this list already appeared in the previous issue of this Newsletter, but several additions and extensions have been performed since then; therefore, the updated list is included here, for completeness. For instance, a new user-callable routine (`IB03BD`), based on a MINPACK-like, structure-exploiting Levenberg-Marquardt algorithm, has been developed. Moreover, `MD03AD` now includes an option for using a fast Cholesky factorization algorithm. Other options allow the use of condition number estimators (instead of checking the diagonal entries, as done in the MINPACK package), or controlled printing of the error norms during the iterative optimization process.

Name	Function
`IB03AD`	to compute a set of parameters for approximating a Wiener system in a least-squares sense, using a neural network approach and a Levenberg-Marquardt algorithm.
`IB03BD`	to compute a set of parameters for approximating a Wiener system in a least-squares sense, using a neural network approach and a MINPACK-like, structure-exploiting Levenberg-Marquardt algorithm.
`MB02WD`	to solve a system of linear equations $Ax = b$, with A symmetric, positive definite, or, in the implicit form, $f(A, x) = b$, where $y = f(A, x)$ is a symmetric positive definite linear mapping from x to y, using a conjugate gradient algorithm without preconditioning.
`MB02XD`	to solve a set of systems of linear equations, $A^T AX = B$, or, in the implicit form, $f(A)X = B$, with $A^T A$ or $f(A)$ positive definite, using symmetric Gaussian elimination.
`MB02YD`	to solve a system of linear equations $Ax = b$, $Dx = 0$, in the least squares sense, with D a diagonal matrix, given a QR factorization with column pivoting of A.

MD03AD	to find the parameters θ for a function $F(x, \theta)$ that give the best approximation for $y = F(x, \theta)$ in a least-squares sense using a Levenberg-Marquardt algorithm based on conjugate gradients or Cholesky factorization for solving linear systems.
MD03BD	to find the parameters θ for a function $F(x, \theta)$ that give the best approximation for $y = F(x, \theta)$ in a least-squares sense using a Levenberg-Marquardt algorithm based on QR factorization with block column pivoting.
MD03BX	to compute the QR factorization with column pivoting of an $m \times n$ matrix J ($m \geq n$), that is, $JP = QR$, where Q is a matrix with orthogonal columns, P a permutation matrix, and R an upper trapezoidal matrix with diagonal elements of nonincreasing magnitude, and to apply the transformation Q^T on the error vector e. The 1-norm of the scaled gradient is also returned.
MD03BY	to find a value for the parameter λ such that if x solves the system $Ax = b$, $\lambda^{1/2} Dx = 0$, in the least squares sense, where A is an $m \times n$ matrix, D is an $n \times n$ nonsingular diagonal matrix, and b is an m-vector, and if δ is a positive number, then either $\lambda = 0$ and $(\|Dx\|_2 - \delta) \leq 0.1\delta$, or $\lambda > 0$ and $\| \|Dx\|_2 - \delta \| \leq 0.1\delta$. It is assumed that a QR factorization with column pivoting of A is available, that is, $AP = QR$, where P is a permutation matrix, Q has orthogonal columns, and R is an upper triangular matrix with diagonal elements of nonincreasing magnitude.
NF01AD	to compute the output of a Wiener system.
NF01AY	to compute the output of a set of neural networks.
NF01BD	to compute the Jacobian of a Wiener system.
NF01BP	to find a value for the parameter λ such that if x solves the system $Jx = b$, $\lambda^{1/2} Dx = 0$, in the least squares sense, where J is an $m \times n$ matrix, D is an $n \times n$ nonsingular diagonal matrix, and b is an m-vector, and if δ is a positive number, then either $\lambda = 0$ and $(\|Dx\|_2 - \delta) \leq 0.1\delta$, or $\lambda > 0$ and $\| \|Dx\|_2 - \delta \| \leq 0.1\delta$. It is assumed that a QR factorization with block column pivoting of J is available, that is, $JP = QR$, where P is a permutation matrix, Q has orthogonal columns, and R is an upper triangular matrix with diagonal elements of nonincreasing magnitude for each block.
NF01BQ	to solve a system of linear equations $Jx = b$, $Dx = 0$, in the least squares sense, with D a diagonal matrix, given a QR factorization with block column pivoting of J.

`NF01BR`	to solve one of the systems of linear equations $Rx = b$, or $R^T x = b$, in the least squares sense, where R is an $n \times n$ block upper triangular matrix, with the structure $$\left[\begin{array}{cccc|c} R_1 & 0 & \cdots & 0 & L_1 \\ 0 & R_2 & \cdots & 0 & L_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & R_\ell & L_\ell \\ 0 & 0 & \cdots & 0 & R_{\ell+1} \end{array}\right],$$ with the upper triangular submatrices R_k, $k = 1\colon \ell+1$, square, and the first ℓ of the same order. The diagonal elements of each block R_k have nonincreasing magnitude. The matrix R is stored in a compressed form.
`NF01BS`	to compute the QR factorization with block column pivoting of an $m \times n$ matrix J $(m \geq n)$, that is, $JP = QR$, where Q is a matrix with orthogonal columns, P a permutation matrix, and R an upper trapezoidal matrix with diagonal elements of nonincreasing magnitude for each block, and to apply the transformation Q^T on the error vector e. The 1-norm of the scaled gradient is also returned.
`NF01BU`	to compute the matrix $J^T J + cI$, for the Jacobian J given in a compressed form.
`NF01BV`	to compute the matrix $J^T J + cI$, for the Jacobian J fully given, for one output variable.
`NF01BW`	to compute the matrix-vector product $x \leftarrow (J^T J + cI)x$, where J is given in a compressed form.
`NF01BX`	to compute $x \leftarrow (A^T A + cI)x$, where A is an $m \times n$ real matrix, and c is a scalar.
`NF01BY`	to compute the Jacobian of the error function for a neural network (for one output variable).
`TB01VD`	to convert the linear discrete-time system given as (A, B, C, D), with initial state x_0, into the output normal form, with parameter vector θ. The matrix A is assumed to be stable. The matrices A, B, C, D and the vector x_0 are transformed, so that on exit they correspond to the system defined by θ.
`TB01VY`	to convert the linear discrete-time system given as its output normal form, with parameter vector θ, into the state-space representation (A, B, C, D), with the initial state x_0.
`TF01MY`	to compute the output sequence of a linear time-invariant open-loop system given by its discrete-time state-space model (A, B, C, D), where A is an $n \times n$ general matrix (the input and output trajectories are stored differently from SLICOT Library routine `TF01MD`).

The Levenberg-Marquardt algorithm using block QR factorization with column pivoting is a specialized, structure-exploiting LAPACK-based implementation of the approach used in the MINPACK package, developed at the Argonne National Laboratory, U.S.A. By a suitable reordering of the parameters describing the Wiener system, the Jacobian matrices (in the

Table 2: The driver routines

Routine	Function
`IB03AD`	Driver routine for identification of the parameter vector for a Wiener system in a least-squares sense, using a neural network approach and conjugate gradients or Cholesky algorithms for solving positive definite systems of linear equations.
`IB03BD`	Driver routine for identification of the parameter vector for a Wiener system in a least-squares sense, using a neural network approach and a MINPACK-like, structure-exploiting Levenberg-Marquardt algorithm.

multi-output case) could be put in a block diagonal form, with an additional block column at the right. This structure is preserved in a QR factorization with column pivoting, if the pivoting is restricted to each block column. This strategy makes sense in the identification context, due to the noise components (usually, the block columns have full rank). The rank deficient case is also covered when solving the associated linear systems. For reliability, an option for finding the ranks by incremental condition estimation is provided.

Several additional lower-level routines have been implemented, but they have not been included in the table above. Also, several MEX-files have been written. The SLICOT toolbox for nonlinear state-space model identification is available since February 2002, but it has been updated in April and June, for the new additions, extensions and improvements. The driver routines of this toolbox are listed in Table 2, and most of the auxiliary routinees are mentioned in the long table above.

All MEX-files for Wiener system identification and related calculations are listed in Table 3.

Table 3: SLICOT Wiener system identification: MEX-file interfaces to MATLAB/Scilab.

MEX-file	Function
`wident`	Computes a discrete-time model of a Wiener system using a neural network approach and a MINPACK-like Levenberg-Marquardt algorithm.
`identc`	Computes a discrete-time model of a Wiener system using a neural network approach and a Levenberg-Marquardt algorithm, based on either a Cholesky, or conjugate gradients algorithm for solving linear systems of equations.
`Wiener`	Computes the output of a Wiener system.
`ldsim`	Computes the output response of a linear discrete-time system (much faster than the MATLAB function lsim).
`onf2ss`	Transforms a linear discrete-time system given in the output normal form to a state-space representation.
`ss2onf`	Transforms a state-space representation of a linear discrete-time system into the output normal form.

The main MEX-files are `wident` and `widentc`, but the remaining MEX-files offer additional working flexibility.

Four MATLAB M-files, which call some of the MEX-files, have been designed to simplify the MEX-files usage, and are presented in Table 4. The *system object* defined in the MATLAB Control Toolbox is used, whenever possible. No M-files corresponding to the MEX-files `wident` and `widentc` are provided, since no simplification of their calling statements is possible.

Table 4: SLICOT Wiener system identification: M-file interfaces.

M-file	Function
`NNout`	Computes the output of a set of neural networks used to model the nonlinear part of a Wiener system.
`dsim`	Computes the output response of a linear discrete-time system (much faster than the MATLAB function lsim).
`o2s`	Transforms a linear discrete-time system given in the output normal form to a state-space representation.
`s2o`	Transforms a state-space representation of a linear discrete-time system into the output normal form.

The SLICOT Wiener identification toolbox has been completed. It has been used (via the M- and MEX-files interfaces) to identity 22 benchmark and industrial examples from the DAISY collection (available at the WEB site `http://www.esat.kuleuven.ac.be/sista/daisy`), on which the linear identification software was also tested. All algorithms have been used, and various options experienced. The algorithms based on Cholesky factorization and QR factorization have been usually faster (and sometimes over 20 times faster) than the algorithms based on the conjugate gradients solver.

The documentation of the use of the developed routines, and their integration into MATLAB and Scilab via MEX-files has been documented in the SLICOT Working Note 2002-6.

Summarizing, the main achievements (for Task III.B) during the last six months are:

1. Implementing new algorithms and interfaces for Wiener systems identification, and developing the associated documentation.

2. Impressive speeding-up the optimization part for identification of Wiener systems, compared with the initial implementation, based on conjugate gradients.

3. Improving the reliability by providing an option for using incremental condition estimation in the Levenberg-Marquardt algorithm based on QR factorization. (This version also ensures scaling invariance.)

4. Improving the functional capabilities; e.g., an option enables to track the sum of squares of the error functions during the iterative optimization process.

5. Additional testing of the NICONET software, including 22 difficult applications from the DAISY collection.

6. Improving the awareness of the industrial community about the capabilities and performances of the SLICOT software: a report and two conference papers have been written.

Vasile Sima and Michel Verhaegen

5 SLICOT tools for robust control

In the last few months, the Sub-Group of robust control have been concentrating on the development of interface with Scilab. Most of the robust control systems analysis and design routines developed in SLICOT would be soon directly callable from Scilab. This interface would enhance the use of SLICOT package.

The Sub-Group have also conducted a major robust design case study over this period, *the robust control of a liquid propellent rocket*. Large uncertainties including the propellant motion in the tanks and varying aerodynamic coefficients made the design problem very challenging. This exercise was tackled using μ-analysis and synthesis method, in both continuous-time and discrete-time cases. Satisfactory results have been obtained. A report on the designs is in preparation, which would be a good tutorial for control engineers and students to consider similar design problems.

In addition to the above, a couple of new routines have been developed which are used in the $\mu-$analysis and synthesis method.

Da-Wei Gu

6 SLICOT tools for nonlinear systems in robotics

6.1 Nonlinear Systems Control

Two reports related to this task were released in March 2002. One of them[1] describes the implemented SLICOT and MATLAB interfaces for the solution of optimization problems, by means of the FSQP package (see issue no. 7, July 2001, of this Newsletter). The other report[2] covers the SLICOT and MATLAB interfaces for the solution of nonlinear algebraic equations systems, by means of the KINSOL package (see also issue no. 7 of this newsletter).

Another report is in preparation that will describe the work done in the optimal control of robot manipulators. As will be explained in the report, to be released in July 2002, two approaches have been explored for the optimal control. The initial approach restricted the possible choices for the weight matrices in order to be able to use a simpler method. In particular, this restriction avoids the necessity to solve a Differential Riccati Equation (DRE), which is formulated as

$$\begin{array}{c} \dot{X} = A_{21} + A_{22}X - XA_{11} - XA_{12}X = F(t,X), \\ X(0) = X_0. \end{array} \tag{1}$$

Later, in a second approach, the restrictions on the weight matrices were removed, and the solution of the DRE was carried out by means of the DRESOL package. The package was modified in order to improve its efficiency and portability, as was described in the last issue of this newsletter.

On the other hand, there has been important progress on the test case for this task, a demonstrator for simulation and optimization of water supply networks, develeloped by UPV and OMRON.

In these last 6 months we have focused on the interconnection between the hydraulic simulator and an OMRON SCADA application, wich will take the test case closer to an industrial real-time control environment. This interconnection will make it possible to simulate the behavior of the system under any control actions, taken from the SCADA application. It will allow to reproduce past scenarios and to compare the model results with the real values of the water distribution network. In this way, it will be possible to make a predictive simulation of the system in order to control different parameters (pressure, flow distribution, cost, water quality,...). This is a first step for the consideration of an automatic control scheme. The work of interconnection is in an advanced state and will soon be finished.

Also in the context of the same test case, there has been progress in the use of KINSOL to carry out the hydraulic simulation. In particular, we have tested the simulation of more complex networks, including valves and pumps. Good results have been obtained in all cases. A report on the demonstrator will be produced and released in July 2002.

Vicente Hernandez, Fernando Alvarruiz and Enrique Arias

[1] *Definition and Implementation of a SLICOT Interface and a* MATLAB *Gateway for the Solution of Nonlinear Programming Problems* . SLICOT Working Note 2002-3. This report is available by anonymous ftp from `wgs.esat.kuleuven.ac.be` in the directory `pub/WGS/REPORTS/SLWN2002-3.ps.Z`.

[2] *Definition and Implementation of a SLICOT Interface and a* MATLAB *Gateway for the Solution of Nonlinear Equations Systems* . SLICOT Working Note 2002-4. This report is available by anonymous ftp from `wgs.esat.kuleuven.ac.be` in the directory `pub/WGS/REPORTS/SLWN2002-4.ps.Z`.

7 SLICOT: a useful tool in industry? Application of SLICOT model reduction routines

This report presents some results on the application of SLICOT model reduction routines [1, 2, 3] for the reduction of large scale heat conduction models with 2D- and 3D geometry which arise in the solution of inverse heat conduction problems. The heat conduction geometries are spatially discretized by finite elements. The inputs of the resulting linear systems are the surface heat fluxes and the outputs are interior temperatures. SLICOT's model reduction routines applied are Balance & Truncate and Balanced Singular Perturbation Approximation, i.e., AB09xx. The functions are called via the MATLAB gateway MEX-file sysred.m. The computations were performed on a desktop PC and SUN-Fire 6800 workstation (single processor). Two application examples are presented and discussed.

7.1 2D example

The first application example is a two-dimensional FEM-model of a 1"-diameter evaporation tube. In this application the outer boundary heat flux varies along the ϕ-coordinate as shown in Fig. 2. In the model the surface heat flux variation is approximated by piecewise linear functions which can be seen in magnification in Fig. 2. Furthermore, temperatures at specific locations $1 - 21$ inside the tube serve as the outputs of the model. A feedback controller which controls a temperature in the tube by adjusting the heat generation rate of the resistance heating coils was added to stabilize the open-loop unstable model.

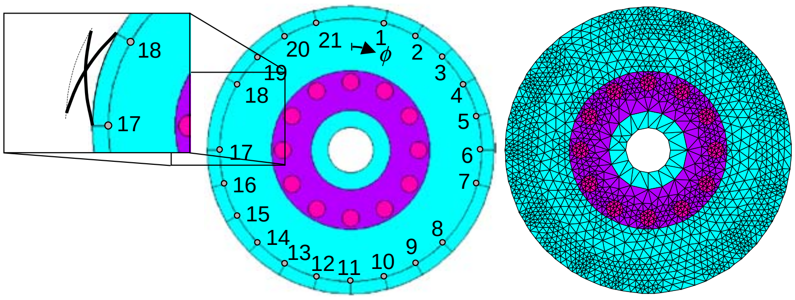

Figure 2: Model of an evaporation tube, linear approximation of boundary heat fluxes, measurement locations (grey dots), left, FEM-discretization (1702 nodes), right.

Model reduction was performed on a PC with an Athlon 1GHz processor and 1GB RAM. The execution time was about 30min. A balanced singular perturbation approximation (SPA) with a prescribed order of $n_r = 21$ was chosen to obtain the reduced model from the original model with state vector dimension of $n = 1702$. Standard tolerances were used. In Fig. 3 the frequency response magnitudes of the original and reduced transfer function are shown for comparison. As can be seen, the approximation at low frequencies up to 10 Hz (in which we were interested) is very good whereas higher frequencies above approx. 50 Hz are not approximated well. It is an inherent feature of the SPA-method to yield a good approximation at low frequencies and to preserve the steady-state gain of the system. A reduced model of higher order than $n_r = 21$ would yield a better approximation towards higher frequencies.

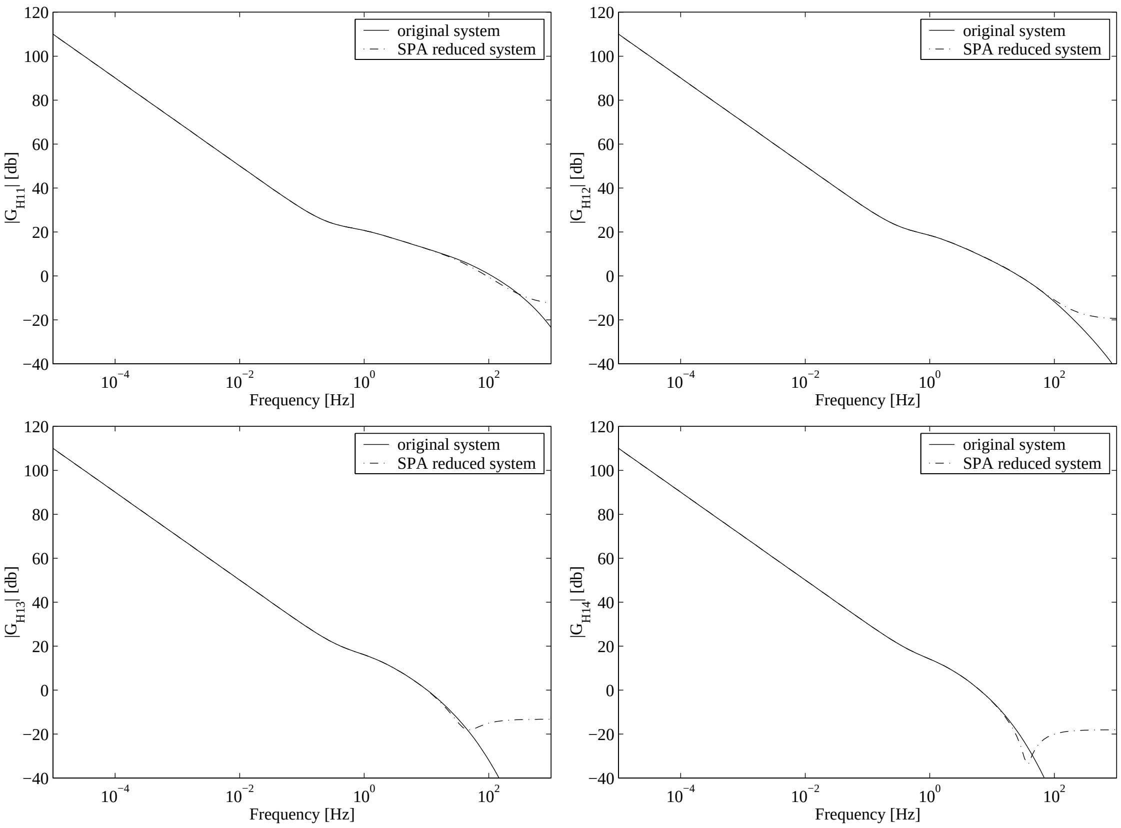

Figure 3: Comparison of the frequency response of the original and reduced order transfer function. Order of the SPA-reduced model is $n_r = 21$.

7.2 3D example

The second example represents a very small section of a tester heater which is used in boiling experiments [4]. The modeled test heater section is approximately 1.2mm × 1.2mm × 5mm in size. A picture of the 3D-FEM-model of the test heater section is shown on the left hand side of Fig. 4. In the model the boundary surface has been divided into 18 separate triangular areas to approximate the heat fluxes by linear varying functions as shown on the right in Fig. 4. Temperatures are measured 10μm (using microthermocouples) below the 15 corner nodes of the each triangle. The discretization of the heater solid model has been performed with tetrahedral elements on a non-uniform grid which is locally refined around the microthermocouple locations (2562 Nodes, 9976 Elements). The non-uniform discretization allows to represent the measurement positions in the model with accuracy while maintaining the size of the model reasonable (the ratio of length scales in the problem is very large, 1.2mm to 10μm).

Finally, a linear state space model with dimension $n = 2562$, $m = 15$ inputs and $p = 15$ outputs is obtained. Due to the integrating behaviour the model is marginally stable.

Using the numerical routines in SLICOT a reduced model with order $n = 161$ was obtained by balancing and truncating the right coprime factorization of the state-space model of the heater. The order of the reduced model was determined automatically. The standard tolerances were used. Balance & Truncation was chosen, because we were interested in a

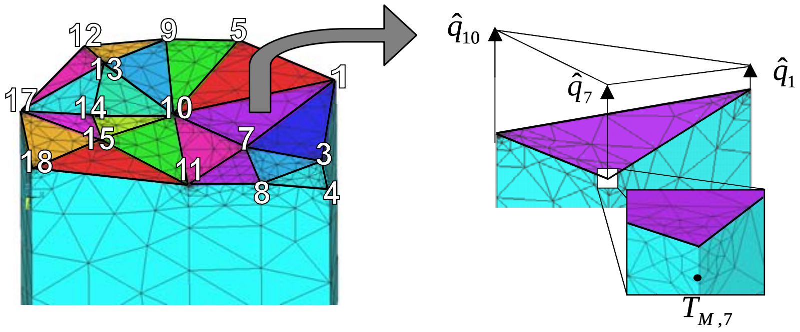

Figure 4: 3D-FEM model of test heater section (2562 Nodes, 9976 Elements), linear approximation of boundary heat fluxes, measurement locations.

good approximation at low and medium frequencies. The reduction has been performed on a SUN-Fire 6800, 900MHz with 2GB of memory in about 2.5h.

Another large model which was recently reduced is of order $n = 4850$ with $p = 18$ inputs and $m = 18$ outputs. It took approximately 19h on a SUN-Fire 6800, 900 Mhz workstation and required 4 GB of RAM. In all cases, the reduced order models approximate the frequency response of the original very well.

7.3 Conclusions

We successfully used the serial versions SLICOT model reduction routines to reduce large linear models obtained after spatial FE-discretization of 2D- and 3D-heat conduction geometries. We presented two successful examples.

It must be pointed out that we also have worked with the μ-analysis toolbox of MATLAB to reduce linear FEM-models. Small models up to the order of a few hundred states can be reduced with the μ-analysis toolbox, but computation of the reduced order model is less efficient. One example is the reduction of a model with a state vector dimension of $n = 300$, four inputs and outputs. The μ-analysis toolbox (sfncfbal.m) needed approx. 2h whereas for SLICOT's sysred.m needed only 4min and yielded even a much better approximation. In the results obtained with the μ-analysis toolbox important system properties such as the steady-state gain of the model were not preserved. Furthermore, approximation of the model at higher frequencies was much worse compared to the results with the routines in SLICOT.

Regarding the computation time of 19h and 4GB of RAM for the reduction of a $n = 4850$, $p = 18$, $m = 18$ model, we believe that a limit is almost reached concerning the problem size of what current well-equipped workstations and direct model reduction methods for dense systems can tackle.

However, there are some applications where a finer discretization would be advantageous in order to resolve higher frequency components in the transfer function of the FEM-model. Reduction of models larger than approx. $n = 6000 - 7000$ states is impraticable, since the computational burden of dense model reduction techniques scales with n^3. For this reason, the problem size sometimes has to be limited either by choosing a smaller domain or a coarser discretization, especially in three dimensional problems, since the order of the model also scales approximately with n^3 regarding the dimension of the geometry. Therefore, parallel [5] as well as iterative reduction methods for sparse systems may be very helpful in the future to reduce even larger models arising from FEM-discretization of PDEs.

References

[1] A. Varga. Model reduction software in the SLICOT library. In B. N. Datta, editor, *Applied and Computational Control, Signals and Circuits*, volume 629 of *The Kluwer International Series in Engineering and Computer Science*, pages 239–282. Kluwer Academic Publishers, Boston, 2001.

[2] A. Varga. Numerical methods and software tools for model reduction. In *Proc. of 1st MATHMOD Conference*, pages 226–230, Wien, 1994.

[3] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. In B. N. Datta, editor, *Applied and Computational Control, Signals and Circuits*, volume 1, chapter 10, pages 499–539. Birkhäuser, Boston, 1999. Also, Technical Report NICONET report 97-3, Working Group on Software WGS, ESAT - Katholieke Universiteit Leuven, 1997.

[4] M. Buchholz, T. Lüttich, H. Auracher, and W. Marquardt. Pool boiling at high heat fluxes (part i): local temperature & heat flux fluctuations at the heater surface. In D. Gorenflo and A. Luke, editors, *Proc. Int. Refrig. Conf. Comm. B1*, Paderborn, October 2001. Session B5.8.

[5] P. Benner, E.S. Quintana-Orti, and G. Quintana-Orti. Parallel algorithms for model reduction of discrete-time systems. Technical Report Report 00 18, Zentrum fuer Technomathematik, Universitaet Bremen, 2000.

7.4 Acknowledgments

We would like to thank Dr. A. Varga, DLR, Oberpfaffenhofen, for his help and support as well as various hints during the installation and application of SLICOT.

Torsten Lüttich, Wolfgang Marquardt
Lehrstuhl für Prozesstechnik, RWTH Aachen, D-52064 Aachen, Germany

8 Order Reduction of Large-Scale Systems. Call for papers: Special Issue of *Linear Algebra and Its Applications*

Order reduction is a common theme within the simulation of complex physical processes. Such simulations often result in very large systems. For example, large systems arise due to accuracy requirements on the spatial discretization of fluids or structures, in the context of lumped-circuit approximations of distributed circuit elements, such as the interconnect or package of VLSI chips, or in simulations of micro-electro-mechanical systems (MEMS), which have both electrical and mechanical components. Dimension reduction is generally required for purposes of expediency and/or storage reduction. Applications include compressed representation, efficient data analysis and feature extraction, real-time analysis, construction of low-order control mechanisms, and many others. Various reduction techniques have been devised, but many of these are described in terms that are discipline-oriented or even application-specific even though they share many common features and origins. This special issue is devoted to exposing the similarities of these approaches, identifying common features, addressing application-specific challenges, and investigating how recent reduction methods for linear systems might be applied to nonlinear problems.

LAA has previously published four special issues devoted to the field of Linear Systems and Control: 1983 (vol. 50), 1989 (vols. 122–124), 1994 (vols. 203–204) and 2002 (to appear). The cross fertilization between numerical linear algebra and linear system theory has been very fruitful. Now, we feel it is time to broaden the scope of these interactions. In the past decade there has been considerable activity in the area of dimension reduction for linear dynamical control systems. However, dimension reduction has a much broader range of application and interpretation. The goals of this special issue are to highlight leading approaches and remaining problems in model reduction for linear system theory, emphasize connections to POD, extend theory and methodology to nonlinear problems, address application-specific techniques.

This special issue will be open to all papers with significant new results in dimension reduction of large systems where either linear algebraic methods play an important role or new tools and problems of linear algebraic nature are presented. Survey papers that illustrate common themes across disciplines and application areas, and especially where Linear Algebra techniques play a central role are highly encouraged. Papers must meet the publication standards of LAA and will be refereed in the usual way.

Areas and topics of interest for this special issue include, but are not limited to:

- Methods and Theory for
  - Linear (time-invariant and time-varying) dynamical systems
  - Descriptor (singular) systems
  - Nonlinear dynamical systems
  - Second-order systems
  - Passive systems
  - Infinite-dimensional systems (e.g., PDE based systems)
- Application-Specific Techniques for

  - Conservative systems (e.g. Molecular Dynamics)
  - Computational fluid dynamics
  - Structural analysis (e.g., condensation or sub-structuring)
  - Micro-electro-mechanical systems (MEMS)
  - Image processing
  - Chemical kinetics

- Low-Order Modeling
  - Proper orthogonal decomposition (POD)
  - Wavelet techniques in dimension reduction
  - Reduced-order modeling of distributed circuit elements

- Low-Order Design
  - Low-order filter design techniques
  - Controller reduction techniques

The deadline for submission of papers is **March 31, 2003**, and the special issue is expected to be published in 2004. Papers should be sent to any of its special editors:

Peter Benner
Institut für Mathematik, MA 4-5
TU Berlin
Strasse des 17. Juni 136
D-10623 Berlin (Germany)
benner@math.tu-berlin.de

Roland W. Freund
Bell Laboratories
Room 2C-525
700 Mountain Avenue
Murray Hill, NJ 07974-0636 (USA)
freund@research.bell-labs.com

Danny C. Sorensen
Dept. Comput. & Appl. Mathematics
Rice University
6100 Main St. – MS 134
Houston, TX 77005-1892 (USA)
sorensen@rice.edu

Andras Varga
Institute of Robotics and Mechatronics
DLR Oberpfaffenhofen
P.O.Box 1116
D-82230 Wessling (Germany)
Andras.Varga@dlr.de

Peter Benner

9 NICONET information corner

This section informs the reader on how to access the SLICOT library, the main product of the NICONET project, and how to retrieve its routines and documentation. Recent updates of the library are also described. In addition, information is provided on the newest NICONET reports, available via the NICONET website or ftp site, as well as information about upcoming workshops/conferences organized by NICONET or with a strong NICONET representation.

Additional information about the NICONET Thematic Network can be found from the NICONET homepage World Wide Web URL

`http://www.win.tue.nl/wgs/niconet.html`

9.1 Electronic Access to the Library

The SLICOT routines can be downloaded from the WGS ftp site,

`ftp://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 operational there. 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 example program, example data, execution results, the associated `.html` file, as well as the source code for the called SLICOT routines.

The entire library is contained in a file, called `slicot.tar.gz`, in the SLICOT root directory `/pub/WGS/SLICOT/`. The following Unix commands should be used for decompressing this file:

```
gzip -d slicot.tar
tar xvf slicot.tar
```

The created subdirectories and their contents is summarized below:

`slicot`	contains the files `libindex.html`, `make.inc`, `makefile`, and the following subdirectories:
`benchmark_data`	contains benchmark data files for Fortran benchmark routines (`.dat`);
`doc`	contains SLICOT documentation files for routines (`.html`);
`examples`	contains SLICOT example programs, data, and results (`.f`, `.dat`, `.res`), and `makefile`, for compiling, linking and executing these programs;
`examples77`	the same contents as in subdirectory `examples`, but the programs are compliant with the Fortran 77 standard (with the `MAX` and/or `MIN` intrinsic functions calls in `PARAMETER` statements removed);

`src`	contains SLICOT source files for routines (`.f`), and `makefile`, for compiling all routines and creating an object library;
`SLTools`	contains MATLAB `.m` files and data `.mat` files;
`SLmex`	contains Fortran source codes for MEX-files (`.f`).

Another, similarly organized file, called `slicotPC.zip`, is available in the SLICOT root directory; it contains the MS-DOS version of the source codes of the SLICOT Library, and can be used on Windows 9x/2000/ME or NT platforms. Included are several source makefiles.

After downloading and decompressing the appropriate SLICOT archive, the user can then browse through the documentation on his local machine, starting from the index file `libindex.html` from `slicot` subdirectory.

9.2 SLICOT Library updates in the period January 2002—June 2002

There have been three major SLICOT Library updates during the period January 2002—June 2002: on February 22, April 9, and June 29. Details are given in the files `Release.Notes` and `Release.History`, located in the directory `pub/WGS/SLICOT/` of the ftp site.

The major *SLICOT Library update on February 22, 2002*, included changes in some routines and addition of about 25 new routines. Most of the changes have been performed to initialize some variables in certain cases. Some of them are related to the optimal workspace length. The updated routines are: `AB01MD`, `AB09HD`, `AB09HX`, `AB09KD`, `BB01AD`, `BB02AD`, `MA02CD`, `MB01PD`, `MB03PY`, `MB03WD`, `SB02RD`, `SB03OD`, `SB10RD`, `TB01LD`, `TB01ZD`, `and TG01ED`. Details are given in the file `Release.History`. Few changes have been also done in the example programs `TAB09MD`, `TAB09ND`, `TAB13MD`, `TBB01AD`, and `TBB02AD`, in three benchmark data files, and in two MEX-files (`syscom` and `findBD`).

Over 20 new user-callable and computational routines for basic control problems, and identification of Wiener systems, have been posted on the SLICOT ftp site on February 22. They include *Identification Routines*, *Mathematical Routines*, and *Transformation Routines*, performing the following main computational tasks:

- compute a set of parameters for approximating a Wiener system in a least-squares sense, using a neural network approach and a Levenberg-Marquardt algorithm.
- solve a system of linear equations $Ax = b$, with A symmetric, positive definite, or, in the implicit form, $f(A, x) = b$, where $y = f(A, x)$ is a symmetric positive definite linear mapping from x to y, using the conjugate gradient algorithm without preconditioning.
- solve a set of systems of linear equations, $A^T AX = B$, or, in the implicit form, $f(A)X = B$, with $A^T A$ or $f(A)$ positive definite, using symmetric Gaussian elimination.
- solve a system of linear equations $Ax = b$, $Dx = 0$, in the least squares sense, with D a diagonal matrix, given a QR factorization with column pivoting of A.
- find the parameters θ for a function $F(x, \theta)$ that give the best approximation for $y = F(x, \theta)$ in a least-squares sense using a Levenberg-Marquardt algorithm based on conjugate gradients for solving linear systems.
- find the parameters θ for a function $F(x, \theta)$ that give the best approximation for $y = F(x, \theta)$ in a least-squares sense using a Levenberg-Marquardt algorithm based on QR factorization with block column pivoting.

- compute the QR factorization with block column pivoting of an $m \times n$ matrix J ($m \geq n$), that is, $JP = QR$, where Q is a matrix with orthogonal columns, P a permutation matrix, and R an upper trapezoidal matrix with diagonal elements of nonincreasing magnitude, and apply the transformation Q^T on the error vector e; the 1-norm of the scaled gradient is also returned.

- find a value for the parameter λ such that if x solves the system $Ax = b$, $\lambda^{1/2}Dx = 0$, in the least squares sense, where A is an $m \times n$ matrix, D is an $n \times n$ nonsingular diagonal matrix, and b is an m-vector, and if δ is a positive number, then either $\lambda = 0$ and $(\|Dx\|_2 - \delta) \leq 0.1\delta$, or $\lambda > 0$ and $|\,\|Dx\|_2 - \delta\,| \leq 0.1\delta$. It is assumed that a QR factorization with column pivoting of A is available, that is, $AP = QR$, where P is a permutation matrix, Q has orthogonal columns, and R is an upper triangular matrix with diagonal elements of nonincreasing magnitude.

- compute the output of a Wiener system.

- compute the output of a set of neural networks.

- compute the Jacobian of a Wiener system.

- find a value for the parameter λ such that if x solves the system $Jx = b$, $\lambda^{1/2}Dx = 0$, in the least squares sense, where J is an $m \times n$ matrix, D is an $n \times n$ nonsingular diagonal matrix, and b is an m-vector, and if δ is a positive number, then either $\lambda = 0$ and $(\|Dx\|_2 - \delta) \leq 0.1\delta$, or $\lambda > 0$ and $|\,\|Dx\|_2 - \delta\,| \leq 0.1\delta$. It is assumed that a QR factorization with block column pivoting of J is available, that is, $JP = QR$, where P is a permutation matrix, Q has orthogonal columns, and R is a block upper triangular matrix with diagonal elements of nonincreasing magnitude for each block.

- solve a system of linear equations $Jx = b$, $Dx = 0$, in the least squares sense, with D a diagonal matrix, given a QR factorization with block column pivoting of J.

- solve one of the systems of linear equations $Rx = b$, or $R^T x = b$, in the least squares sense, where R is an $n \times n$ block upper triangular matrix, with the structure

$$\left[\begin{array}{cccc|c} R_1 & 0 & \cdots & 0 & L_1 \\ 0 & R_2 & \cdots & 0 & L_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & R_\ell & L_\ell \\ 0 & 0 & \cdots & 0 & R_{\ell+1} \end{array}\right],$$

  with the upper triangular submatrices R_k, $k = 1\colon \ell+1$, square, and the first ℓ of the same order. The diagonal elements of each block R_k have nonincreasing magnitude. The matrix R is stored in a compressed form.

- compute the QR factorization with block column pivoting of an $m \times n$ matrix J ($m \geq n$), that is, $JP = QR$, where Q is a matrix with orthogonal columns, P a permutation matrix, and R a block upper trapezoidal matrix with diagonal elements of nonincreasing magnitude for each block, and apply the transformation Q^T on the error vector e; the 1-norm of the scaled gradient is also returned.

- compute the matrix $J^T J + cI$, for the Jacobian J given in a compressed form.
- compute the matrix $J^T J + cI$, for the Jacobian J fully given, for one output variable.
- compute the matrix-vector product $x \leftarrow (J^T J + cI)x$, where J is given in a compressed form.
- compute $x \leftarrow (A^T A + cI)x$, where A is an $m \times n$ real matrix, and c is a scalar.
- compute the Jacobian of the error function for a neural network (for one output variable).
- convert the linear discrete-time system given as (A, B, C, D), with initial state x_0, into the output normal form, with parameter vector θ. The matrix A is assumed to be stable. The matrices A, B, C, D and the vector x_0 are transformed, so that on exit they correspond to the system defined by θ.
- convert the linear discrete-time system given as its output normal form, with parameter vector θ, into the state-space representation (A, B, C, D), with the initial state x_0.
- compute the output sequence of a linear time-invariant open-loop system given by its discrete-time state-space model (A, B, C, D), where A is an $n \times n$ general matrix (the input and output trajectories are stored differently from SLICOT Library routine `TF01MD`).

All test (example) programs which contained `MAX` and/or `MIN` intrinsic functions calls in `PARAMETER` statements have now a version without these calls, in order to be compliant with the Fortran 77 standard. The modified files (over 100), and all the other example programs (`.f`), data (`.dat`) and results (`.res`) files, are stored in the subdirectory `examples77`.

The MATLAB 5.3 toolboxes have been saved in a new subdirectory, called `SLToolboxes5`, of the `MatlabTools` directory of the ftp site. The former subdirectory `SLToolboxes` now contains the MATLAB 6 versions of all files, including `.dll` files. The MATLAB 5.3 files will not be updated in the future.

A new MEX-file and associated M-file for generating benchmarck examples for algebraic Riccati equations have been made available.

The *SLICOT Library update on April 9, 2002*, included corrections in few routines (`AG08BD`, `AG08BY`, `IB03BD`, `MB04VX`, `MD03BF`, `SB03MX`, `SB03OD`, `SG03BD`, `TB01UD`, `TB01VD`, and `TB01VY`), an example program (`TAG08BD`), four M-files, and six `.html` files. Details are given in the file `Release.History`.

Ten new or updated routines, belonging to the chapters *Identification Routines*, *Mathematical Routines* and *Nonlinear Systems*, have been posted on the SLICOT ftp site. These routines have the following main functionality:

- compute a set of parameters for approximating a Wiener system in a least-squares sense, using a neural network approach and a conjugate gradients or Cholesky-based Levenberg-Marquardt algorithm.
- find the parameters theta for a function $F(x, \theta)$ that give the best approximation for $y = F(x, \theta)$ in a least-squares sense using a Levenberg-Marquardt algorithm based on conjugate gradients or Cholesky factorization for solving linear systems.

- evaluate the functions and Jacobian matrices for optimizing the parameters of the nonlinear part of a Wiener system (initialization phase).

- evaluate the functions and Jacobian matrices for solving a standard nonlinear least squares problem using conjugate gradients or Cholesky-based solvers.

- compute the matrix $J^T J + cI$, for the Jacobian J either given in a compressed form, or fully given (for one output variable).

- compute the matrix-vector product $x \leftarrow (J^T J + cI)x$, for the matrix J either given in a compressed form, or fully given, where c is a scalar.

Three new test programs, a MEX-file and an M-file, covering the above functionality, have been included. The toolbox for nonlinear Wiener systems identification has been updated and completed.

A new directory (of the SLICOT ftp site root directory), `plicmr`, and its subdirectory, `doc`, now contain the on-line `.html` documentation files for the currently available parallel SLICOT library routines (for large order model reduction).

The *SLICOT Library update on June 29, 2002*, included corrections in the routines `AB13FD`, `IB03AD`, `IB03BD`, `MB03NY`, `MC01OD`, `MC01PD`, and `SB02OD`. Details are given in the file `Release.Notes`.

Several new routines belonging to the chapters *Mathematical Routines*, *Synthesis Routines*, and *Transformation Systems*, have been posted on the SLICOT ftp site. These routines have the following main functionality:

- compute the coefficients of a real polynomial $P(x)$ from its zeros. The coefficients are stored in decreasing order of the powers of x.

- solve either the continuous-time or discrete-time algebraic Riccati equations for descriptor systems.

- compute the transfer function matrix G of a state-space representation (A, B, C, D) of a linear time-invariant multivariable system, using the pole-zeros method. Each element of the transfer function matrix is returned in a cancelled, minimal form, with numerator and denominator polynomials stored either in increasing or decreasing order of the powers of the indeterminate.

- separate the strictly proper part from the constant part of a proper transfer function matrix.

- compute the sum of an p-by-m rational matrix and a real p-by-m matrix.

- compute the gain of a single-input single-output linear system, given its state-space representation (A, b, c, d), and its poles and zeros. The matrix A is assumed to be in an upper Hessenberg form.

- compute the transfer function matrix G of a state-space representation (A, B, C, D) of a linear time-invariant multivariable system, using the pole-zeros method. The transfer function matrix is returned in a minimal pole-zero-gain form.

Three associated example programs have been written.

In addition, two new MEX-files, performing (partial) pole assignment, and computing the periodic Hessenberg or periodic Schur decomposition of a matrix product, respectively, and four associated M-files have been made available on the SLICOT ftp site.

The files describing the Tasks I.A, III.A, III.B, and IV.A have been updated.

9.3 New NICONET Reports

Recent NICONET reports (available after January 2002), that can be downloaded as compressed postscript files from the World Wide Web URL

`http://www.win.tue.nl/wgs/reports.html`

or from the WGS ftp site,

`ftp://wgs.esat.kuleuven.ac.be`

(directory `pub/WGS/REPORTS/`), are the following:

- Peter Benner, Enrique S. Quintana-Orti, Gregorio Quintana-Orti, Rafael Mayo. *Enhanced Services for Remote Model Reduction of Large-Scale Dense Linear Systems* (file `SLWN2002-1.ps.Z`, January 2002).

  This paper describes enhanced services for remote model reduction of large-scale, dense linear time-invariant systems. Specifically, we describe a mail service and a web service for model reduction on a cluster of Intel Pentium-II architectures using absolute and relative error methods. Experimental results show the appeal and accessibility provided by these services.

- Y. Chahlaoui and P. Van Dooren. *A collection of Benchmark examples for model reduction of linear time invariant dynamical systems* (file `SLWN2002-2.ps.Z`, February 2002 and revised in March 2002).

  In order to test the numerical methods for model reduction we present here a benchmark collection, which contain some useful real world examples reflecting current problems in applications. All simulations were obtained via Matlab and some SLICOT programs of NICONET.

- F. Alvarruiz and V. Hernandez. *Definition and implementation of a SLICOT interface and a* MATLAB *Gateway for the solution of non-linear programming problems* (file `SLWN2002-3.ps.Z`, March 2002).

  This paper presents SLICOT and MATLAB interfaces for the FSQP package, which stands for Feasible Sequential Quadratic Programming. The SLICOT interface enables the user to call the FSQP package by means of a subroutine with a SLICOT- compliant calling sequence. By means of the MATLAB interface the user can call the package from MATLAB, defining the problem by means of MATLAB functions. The interfaces could be extended in the future in order to consider other nonlinear programming solvers, although some restructuring of the interfaces would be necessary.

- F. Alvarruiz and V. Hernandez *Definition and implementation of a SLICOT interface and a* MATLAB *Gateway for the solution of nonlinear equations systems* (file `SLWN2002-4.ps.Z`, March 2002).

  This paper presents SLICOT and MATLAB interfaces for the KINSOL software package, used for solving nonlinear equations systems. The SLICOT interface enables the user to call the KINSOL package by means of a subroutine with a SLICOT-compliant calling sequence. By means of the MATLAB interface the user can call the package from MATLAB, defining the problem by means of MATLAB functions. The interfaces could be extended in the future in orde to consider other nonlinear equations systems solvers, although some restructuring of the interfaces would be necessary.

- Rene Schneider, Andreas Riedel, Vincent Verdult, Michel Verhaegen, Vasile Sima. *SLICOT System Identification Toolbox for Nonlinear Wiener Systems* (file `SLWN2002-6.ps.Z`, June 2002).

  This report summarizes the achievements and deliverables of the Task III.B of the NICONET Project. After a short description of the nonlinear Wiener system identification problem, the numerical algorithms implemented in the SLICOT Nonlinear Systems Identification Toolbox are surveyed. The associated Fortran routines are then listed and their functional abilities are outlined. The developed interfaces to MATLAB or Scilab, as well as examples of use are presented.

Previous NICONET/WGS reports are also posted at the same address.

9.4 Forthcoming Conferences

Forthcoming Conferences related to the NICONET areas of interest, where NICONET partners submitted proposals for NICONET/SLICOT-related talks and papers, and disseminated or will disseminate information and promote SLICOT, include the following:

- IFAC Congress, Barcelona, July 2002.
- SIAM's 50th Anniversary and 2002 Annual Meeting, Philadelphia Marriott Hotel, July 8-12, 2002.
- MTNS, "Mathematical Theory of networks and Systems" meeting 2002, University of Notre Dame, South Bend, Indiana, USA, August 12-16, 2002, see `http://www.nd.edu/~mtns/`
- Joint "IEEE Conference on Control Applications" and "IEEE Conference on Computer Aided Control Systems Design", September 17-20, 2002, Scottish Exhibition & Conference Centre, Glasgow, Scotland.

Vasile Sima