HP 9000 Systems

HP MLIB V4.1 Release Note

HP Part No. B5649-90008
Printed in U.S.A.
E0697

Edition 1

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

HP MLIB V4.1 Exemplar Scalable Servers Release Note

Contents

HP MLIB V4.1 Exemplar Scalable Servers Release Note

This document describes HP MLIB V4.1 for Hewlett-Packard Exemplar scalable technical servers. This document is intended to provide HP MLIB users with information not readily available in the HP MLIB User's Guides provided with this release. Always refer to this release note before reporting questions or problems with HP MLIB; your questions may be answered here.

This document contains the following sections:

- Introduction
- Contents
- Documentation
- Directories and Files
- Using HP MLIB
- New Features
- Parallelized Subprograms
- Known Documentation Problems

Introduction

HP MLIB V4.1 is a package of high-performance mathematical software in the form of subroutine libraries. The MLIB product is designed for use by software engineers who develop, convert, or optimize engineering or scientific programs that rely heavily on computational kernels such as the Basic Linear Algebra Subprograms, linear algebra computations such as solving linear equations or eigenvalue problems, discrete Fourier transforms, or convolutions. Using MLIB will save you the time and effort of developing, testing, or optimizing these algorithms.

MLIB versions are available for HP Exemplar SPP1200 and SPP1600 systems, and for C-, D-, K-, S-, and X-Class systems. These are specifically optimized for each system type but are usage-compatible, promoting portability while retaining efficiency across HP's workstation and server lines. This release supports SPP1200, SPP1600, S-Class, and X-Class systems with the following libraries tuned for each system type:

- VECLIB (*libveclib.a*)
- SCILIB (*libscilib.a*)
- LAPACK (*liblapack.a*)

Subprograms that take advantage of Exemplar's parallel features are listed in Table 1 on page 10, Table 2 on page 11, and Table 3 on page 12.

Contents

The software distribution package for HP MLIB V4.1 consists of:

- This Release Note
- Distribution media for the software

Documentation

HP MLIB V4.1 is documented in the following locations:

- This Release Note (B5649-90008)
- *HP MLIB LAPACK User's Guide*, Third Edition (B5649-90001)
- *HP MLIB SCILIB User's Guide*, Second Edition (B5649-90002)
- *HP MLIB VECLIB User's Guide*, Second Edition (B5649-90003)
- The MLIB Man Pages

Directories and Files

HP MLIB V4.1 includes three library files for HP Exemplar SPP1200 and SPP1600 systems, which are based on the PA1.1 architecture, and three for HP Exemplar S-Class and X-Class systems, which are based on the PA2.0 architecture. The library file names for all four system types are *liblapack.a*, *libscilib.a*, and *libveclib.a*. The SPP1200 and SPP1600 versions of the libraries are installed in directory /opt/mlib/lib/pa1.1parallel, while the S-Class and X-Class versions are installed in /opt/mlib/lib/pa2.0parallel. See "Using HP MLIB" for instructions on accessing the libraries.

Three header files, *lapack.h*, *scilib.h*, and *veclib.h*, for use with a C or C++ compiler, are installed in /opt/mlib/include.

Man pages for HP MLIB are located in section 3m of the on-line Programmer's Reference Manual; they are installed in the /opt/mlib/share/man directory. Man pages are available for each chapter introduction and every subprogram documented in the following books, shipped with this release:

- *HP MLIB LAPACK User's Guide*
- *HP MLIB SCILIB User's Guide*
- *HP MLIB VECLIB User's Guide*

You must add the directory /opt/mlib/share/man to your MANPATH shell environment variable to access the associated man pages.

An on-line copy of this release note is installed in /opt/mlib/newconfig/RelNotes.

Using HP MLIB

Linking Your Program With HP MLIB V4.1

HP MLIB V4.1 subprograms must be loaded by the Fortran 77 or Fortran 90 compiler, and the resulting executable must be executed under the SPP-UX 5.2 operating system.

There are several ways to specify linking your program with LAPACK, SCILIB, or VECLIB:

1. Specify the entire path of the library file on the f77 or f90 command line that links your program. For example, to use VECLIB with the f77 compiler on an S-Class system, use

   f77 [*options*] *file* **/opt/mlib/lib/pa2.0parallel/libveclib.a**

2. Use the –l option on the f77 or f90 command line that links your program, preceded by either **–Wl,–L /opt/mlib/lib/pa1.1parallel** or **–Wl,–L/opt/mlib/lib/pa2.0parallel**. For example, the above f77 command line could be written as:

   f77 [*options*] *file* **–Wl,–L/opt/mlib/lib/pa2.0parallel –lveclib**

3. Set your LDOPTS environment variable to include the appropriate one of **–L/opt/mlib/lib/pa1.1parallel** or **–L/opt/mlib/lib/pa2.0parallel**, and use the –l option on the f77 or f90 command line that links your program. For example, with the f90 compiler, use

   f90 [*options*] *file* **–lveclib**

Parallel Processing

Parallel processing is available on Hewlett-Packard Exemplar scalable technical servers. You can permit or disable parallel processing at link time or at run time. A program will not use parallelism in MLIB unless it is enabled both at link time and at run time.

Linking for Parallel or Non-Parallel Processing

To permit parallel processing at link time, your link step must produce a multithreaded executable. You always get a multithreaded executable if you link with the Fortran 77 or C compiler using the **+Oparallel** flag:

f77 [*options*] *file* **+O3 +Oparallel –lveclib**

With Fortran 90, you must include the **–Wl,+tm,S2000,+parallel** option on the f90 command line that links your program:

f90 [*options*] *file* **–Wl,+tm,S2000,+parallel –lveclib –lpthread –lcps –lpthread –lail**

To disable automatic parallelism at link time, you omit the **+Oparallel** or **+parallel** option and include several runtime libraries if they are needed. The following commands are the non-parallelizing equivalents of the previous parallelizing linking commands.

f77 [*options*] *file* **–lveclib –lpthread –lcps –lpthread –lail**

f90 [*options*] *file* **–Wl,+tm,S2000 –lveclib –lpthread –lcps –lpthread –lail**

Another alternative is to link with parallelism enabled and use the *mpa*(1) utility to modify the attributes of the executable file to disable parallelism. Refer to the *mpa*(1) man page for details.

Controlling Parallelism at Run Time

Each parallelized subprogram in previous MLIB releases automatically used parallel processing when it determined that parallelism was appropriate. The decision was based on such factors as whether multiple processors were available, whether the program already was using multiple threads, and the problem size. This default action caused problems for users who wanted to control parallelism via other means, such as the Message Passing Interface, MPI. *Therefore, HP MLIB V4.1 makes a radical change in the way parallelism is controlled within MLIB subprograms.*

The biggest change in parallelism control is that MLIB subroutines do not invoke parallel processing by default. If you simply compile and link your

program, parallelized MLIB subprograms will execute on a single processor even though the executable is marked for parallel execution.

To retain parallel capability within HP MLIB, two methods were added to specify the extent of parallel processing to be allowed. First, a shell environment variable, MLIB_NUMBER_OF_THREADS, allows you to enable parallelism within MLIB subprograms and to specify the maximum number of threads that may be used in parallel regions. Not setting MLIB_NUMBER_OF_THREADS has the same result as setting it to 1 -- parallel processing is disabled within MLIB subroutines. Setting it to a value greater than the number of threads in the subcomplex allows parallelized MLIB subprograms to use all of the CPUs in the subcomplex.

The following command lines show the C shell syntax and Korn shell syntax to use when setting the variable to 8 processors:

```
C shell:      setenv MLIB_NUMBER_OF_THREADS 8
Korn shell:   export MLIB_NUMBER_OF_THREADS=8
```

MLIB_NUMBER_OF_THREADS is examined upon the first call to a parallelized MLIB subprogram to establish the default parallel action within MLIB.

The second method you can use to control parallelism within MLIB is the new subroutine MLIB_SETNUMTHREADS. You can call this subroutine at any time to set the maximum number of parallel threads used in subsequent MLIB calls. The specified value overrides the absence of the MLIB_NUMBER_OF_THREADS environment variable or any value assigned to it. Additionally, you can use MLIB_SETNUMTHREADS to restore MLIB parallel processing to its run-time default. Refer to the *mlib_setnumthreads*(3m) man page for usage information.

Finally, in addition to the above MLIB controls, at run time you can use the *mpa*(1) utility or the MP_NUMBER_OF_THREADS environment variable to control parallelism. These controls set the maximum amount of parallelism that your program will use, and the MLIB-specific mechanisms offer finer control within that maximum. Refer to the *mpa*(1) or *f77*(1) man page, respectively, for details.

New Functionality

The sparse linear equation solver, described in Chapter 6 of the HP MLIB VECLIB User's Guide and the *sparse-solver*(3m) man page, has been parallelized for SPP1200 and SPP1600 systems, and several new subprograms have been added to give the user additional control over the package or additional output from it. These are documented in the following new man pages: *dsleff*(3m), *dslemp*(3m), *dsleop*(3m), and *dslerd*(3m). This functionality was provided in the HP MLIB V4.0 release for S-Class systems.

This HP MLIB release includes subprograms that compute tapered convolutions. These are documented in the *stconv*(3m) and *dtconv*(3m) man pages.

CPUTIME and WALLTIME are utility subprograms included in the VECLIB and LAPACK libraries. They make it easy to measure intervals of CPU time and wallclock time, respectively, in a program. See the *cputime*(3m) and *walltime*(3m) man pages for details.

HP MLIB V4.1 includes the SCILIB library. SCILIB is a collection of Fortran-callable mathematical subprograms which provides a look-alike implementation of the Scientific Library portion of Cray Research's UNICOS

Math and Scientific Library, V5.0. SCILIB had been provided as part of ConvexMLIB V3.0 for SPP1200 and SPP1600 systems, but is new to S- and X-Class systems.

Parallelized Subprograms

Tables 1-3 list the name, chapter of the user's guide, and purpose of various subroutines in LAPACK, SCILIB, and VECLIB that have been parallelized to improve performance on Exemplar servers. Many additional subprograms in the libraries call the BLAS, and will inherit their parallelism.

Table 1: Parallelized Subprograms in LAPACK

Subroutine name	Chapter	Description
{S,D,C,Z}GBSV	2	Solve a General Band Linear System
{S,D,C,Z}GESV	2	Solve a General Full Linear System
{S,D,C,Z}GBSVX	3	Solve a General Band Linear System
{S,D,C,Z}GESVX	3	Solve a General Full Linear System
S,D,C,Z}GBTRF	4	Factor a General Band Matrix
{S,D,C,Z}GETRF	4	Factor a General Full Matrix
{S,D,C,Z}GETRI	4	Invert General Matrix
{S,D,C,Z}GETRS	4	Solve a General Full Linear System
{S,D}GELS	5	Solve General Least Squares Problem
{S,D}GELSS	5	Solve General Least Squares Problem
{S,D}GEQRF	6	QR Factorization of a General Matrix
{S,D}GGQRF	6	Generalized QR Factorization
{S,D}GGRQF	6	Generalized RQ Factorization
{S,D}GEGS	9	Generalized Schur Form
{S,D}GEGV	9	Generalized General Matrix Eigenproblem
{S,D}GESVD	10	Singular Value Decomposition
{S,D,C,Z}GEMM	-	General Matrix-Matrix Multiply
{S,D,C,Z}GEMV	-	General Matrix-Vector Multiply
{S,D}GER	-	General Rank-1 Update
{C,Z}GERC	-	General Rank-1 Update
{C,Z}GERC	-	General Rank-1 Update
{C,Z}GERU	-	General Rank-1 Update

Table 2: Parallelized Subprograms in SCILIB

Subroutine name	Chapter	Description
CFFT2	6	Complex-to-Complex One-Dimensional FFT
CFFTMLT	6	Simultaneous Complex-to-Complex One-Dimensional FFT
CRFFT2	6	Complex-to-Real One-Dimensional FFT
RCFFT2	6	Real-to-Complex One-Dimensional FFT
RFFTMLT	6	Simultaneous Real-to-Complex One-Dimensional FFT

Table 3: Parallelized Subprograms in VECLIB

Subroutine name	Chapter	Description
{S,D,C,Z}GEMM	3	General Matrix-Matrix Multiply
{S,D,C,Z}GEMV	3	General Matrix-Vector Multiply
{S,D}GER	3	General Rank-1 Update
{C,Z}GERC	3	General Rank-1 Update
{C,Z}GERU	3	General Rank-1 Update
{S,D,C,Z}GBFA	4	Factor a General Band Matrix
{S,D}GEFA	4	Factor a General Matrix
{S,D}GESL	4	Solve General Linear Equations
DSLEFA	6	Numeric Factor Sparse Matrix
DSLEFS	6	Solve Sparse Linear Equations
DSKYFA	8	Numeric Factor Skyline Matrix
DSKYFS	8	Solve Skyline Linear Equations
DSKYFX	8	Solve Skyline Linear Equations
{S,D,C,Z}1DFFT	9	One-dimensional FFT
{S,D,C,Z}2DFFT	9	Two-dimensional FFT
{S,D,C,Z}3DFFT	9	Three-dimensional FFT
{S,D,C,Z}FFTS	9d	Simultaneous Complex-to-Complex One-Dimensional FFT
{S,D,C,Z}R12FT	9	Real-to-Complex One-Dimensional FFT
{S,D,C,Z}RC2FT	9	Real-to-Complex Two-Dimensional FFT
{S,D,C,Z}RC3FT	9	Real-to-Complex Three-Dimensional FFT
{S,D,C,Z}RCFTS	9	Simultaneous Real-to-Complex One-Dimensional FFT
{S,D}CONV	10	Correlation/Convolution

Known Documentation Problems

As HP MLIB V4.1 is accompanied with the same user's guides as V4.0, the description of parallel processing at run time is incorrect in all three printed manuals. Please refer to "Controlling Parallelism at Run Time" in this Release Note or one of the man pages *lapack*(3m), *scilib*(3m), or *veclib*(3m) to learn how you can control HP MLIB parallelism at run time.

Contrary to what is written about profiling MLIB applications in Chapter 1 of each of the three HP MLIB User's Guides, MLIB libraries are not instrumented for profiling with CXpa. CXpa is an optional product. If you have it, you can use CXoi, the Exemplar PA-RISC Object and Archive File Instrumentor, to make instrumented MLIB libraries. See *CXoi*(1) for details.

HP 9000 Systems

HP MLIB V4.1 Release Note

HP Part No. B5649-90008
Printed in U.S.A.
E0697

Edition 1

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

HP MLIB V4.1 Exemplar Scalable Servers Release Note

Contents

HP MLIB V4.1 Exemplar Scalable Servers Release Note

This document describes HP MLIB V4.1 for Hewlett-Packard Exemplar scalable technical servers. This document is intended to provide HP MLIB users with information not readily available in the HP MLIB User's Guides provided with this release. Always refer to this release note before reporting questions or problems with HP MLIB; your questions may be answered here.

This document contains the following sections:

- Introduction
- Contents
- Documentation
- Directories and Files
- Using HP MLIB
- New Features
- Parallelized Subprograms
- Known Documentation Problems

Introduction

HP MLIB V4.1 is a package of high-performance mathematical software in the form of subroutine libraries. The MLIB product is designed for use by software engineers who develop, convert, or optimize engineering or scientific programs that rely heavily on computational kernels such as the Basic Linear Algebra Subprograms, linear algebra computations such as solving linear equations or eigenvalue problems, discrete Fourier transforms, or convolutions. Using MLIB will save you the time and effort of developing, testing, or optimizing these algorithms.

MLIB versions are available for HP Exemplar SPP1200 and SPP1600 systems, and for C-, D-, K-, S-, and X-Class systems. These are specifically optimized for each system type but are usage-compatible, promoting portability while retaining efficiency across HP's workstation and server lines. This release supports SPP1200, SPP1600, S-Class, and X-Class systems with the following libraries tuned for each system type:

- VECLIB (*libveclib.a*)
- SCILIB (*libscilib.a*)
- LAPACK (*liblapack.a*)

Subprograms that take advantage of Exemplar's parallel features are listed in Table 1 on page 10, Table 2 on page 11, and Table 3 on page 12.

Contents

The software distribution package for HP MLIB V4.1 consists of:

- This Release Note
- Distribution media for the software

Documentation

HP MLIB V4.1 is documented in the following locations:

- This Release Note (B5649-90008)
- *HP MLIB LAPACK User's Guide,* Third Edition (B5649-90001)
- *HP MLIB SCILIB User's Guide,* Second Edition (B5649-90002)
- *HP MLIB VECLIB User's Guide,* Second Edition (B5649-90003)
- The MLIB Man Pages

Directories and Files

HP MLIB V4.1 includes three library files for HP Exemplar SPP1200 and SPP1600 systems, which are based on the PA1.1 architecture, and three for HP Exemplar S-Class and X-Class systems, which are based on the PA2.0 architecture. The library file names for all four system types are *liblapack.a, libscilib.a,* and *libveclib.a.* The SPP1200 and SPP1600 versions of the libraries are installed in directory /opt/mlib/lib/pa1.1parallel, while the S-Class and X-Class versions are installed in /opt/mlib/lib/pa2.0parallel. See "Using HP MLIB" for instructions on accessing the libraries.

Three header files, *lapack.h, scilib.h,* and *veclib.h,* for use with a C or C++ compiler, are installed in /opt/mlib/include.

Man pages for HP MLIB are located in section 3m of the on-line Programmer's Reference Manual; they are installed in the /opt/mlib/share/man directory. Man pages are available for each chapter introduction and every subprogram documented in the following books, shipped with this release:

- *HP MLIB LAPACK User's Guide*
- *HP MLIB SCILIB User's Guide*
- *HP MLIB VECLIB User's Guide*

You must add the directory /opt/mlib/share/man to your MANPATH shell environment variable to access the associated man pages.

An on-line copy of this release note is installed in /opt/mlib/newconfig/RelNotes.

Using HP MLIB

Linking Your Program With HP MLIB V4.1

HP MLIB V4.1 subprograms must be loaded by the Fortran 77 or Fortran 90 compiler, and the resulting executable must be executed under the SPP-UX 5.2 operating system.

There are several ways to specify linking your program with LAPACK, SCILIB, or VECLIB:

1. Specify the entire path of the library file on the f77 or f90 command line that links your program. For example, to use VECLIB with the f77 compiler on an S-Class system, use

   f77 [*options*] *file* **/opt/mlib/lib/pa2.0parallel/libveclib.a**

2. Use the –l option on the f77 or f90 command line that links your program, preceded by either **–Wl,–L /opt/mlib/lib/pa1.1parallel** or **–Wl,–L/opt/mlib/lib/pa2.0parallel**. For example, the above f77 command line could be written as:

   f77 [*options*] *file* **–Wl,–L/opt/mlib/lib/pa2.0parallel –lveclib**

3. Set your LDOPTS environment variable to include the appropriate one of **–L/opt/mlib/lib/pa1.1parallel** or **–L/opt/mlib/lib/pa2.0parallel**, and use the **–l** option on the f77 or f90 command line that links your program. For example, with the f90 compiler, use

   f90 [*options*] *file* **–lveclib**

Parallel Processing

Parallel processing is available on Hewlett-Packard Exemplar scalable technical servers. You can permit or disable parallel processing at link time or at run time. A program will not use parallelism in MLIB unless it is enabled both at link time and at run time.

Linking for Parallel or Non-Parallel Processing

To permit parallel processing at link time, your link step must produce a multithreaded executable. You always get a multithreaded executable if you link with the Fortran 77 or C compiler using the **+Oparallel** flag:

f77 [*options*] *file* **+O3 +Oparallel –lveclib**

With Fortran 90, you must include the **–Wl,+tm,S2000,+parallel** option on the f90 command line that links your program:

f90 [*options*] *file* **–Wl,+tm,S2000,+parallel –lveclib –lpthread –lcps –lpthread –lail**

To disable automatic parallelism at link time, you omit the **+Oparallel** or **+parallel** option and include several runtime libraries if they are needed. The following commands are the non-parallelizing equivalents of the previous parallelizing linking commands.

f77 [*options*] *file* **–lveclib –lpthread –lcps –lpthread –lail**

f90 [*options*] *file* **–Wl,+tm,S2000 –lveclib –lpthread –lcps –lpthread –lail**

Another alternative is to link with parallelism enabled and use the *mpa*(1) utility to modify the attributes of the executable file to disable parallelism. Refer to the *mpa*(1) man page for details.

Controlling Parallelism at Run Time

Each parallelized subprogram in previous MLIB releases automatically used parallel processing when it determined that parallelism was appropriate. The decision was based on such factors as whether multiple processors were available, whether the program already was using multiple threads, and the problem size. This default action caused problems for users who wanted to control parallelism via other means, such as the Message Passing Interface, MPI. *Therefore, HP MLIB V4.1 makes a radical change in the way parallelism is controlled within MLIB subprograms.*

The biggest change in parallelism control is that MLIB subroutines do not invoke parallel processing by default. If you simply compile and link your

program, parallelized MLIB subprograms will execute on a single processor even though the executable is marked for parallel execution.

To retain parallel capability within HP MLIB, two methods were added to specify the extent of parallel processing to be allowed. First, a shell environment variable, MLIB_NUMBER_OF_THREADS, allows you to enable parallelism within MLIB subprograms and to specify the maximum number of threads that may be used in parallel regions. Not setting MLIB_NUMBER_OF_THREADS has the same result as setting it to 1 -- parallel processing is disabled within MLIB subroutines. Setting it to a value greater than the number of threads in the subcomplex allows parallelized MLIB subprograms to use all of the CPUs in the subcomplex.

The following command lines show the C shell syntax and Korn shell syntax to use when setting the variable to 8 processors:

```
C shell:      setenv MLIB_NUMBER_OF_THREADS 8
Korn shell:   export MLIB_NUMBER_OF_THREADS=8
```

MLIB_NUMBER_OF_THREADS is examined upon the first call to a parallelized MLIB subprogram to establish the default parallel action within MLIB.

The second method you can use to control parallelism within MLIB is the new subroutine MLIB_SETNUMTHREADS. You can call this subroutine at any time to set the maximum number of parallel threads used in subsequent MLIB calls. The specified value overrides the absence of the MLIB_NUMBER_OF_THREADS environment variable or any value assigned to it. Additionally, you can use MLIB_SETNUMTHREADS to restore MLIB parallel processing to its run-time default. Refer to the *mlib_setnumthreads*(3m) man page for usage information.

Finally, in addition to the above MLIB controls, at run time you can use the *mpa*(1) utility or the MP_NUMBER_OF_THREADS environment variable to control parallelism. These controls set the maximum amount of parallelism that your program will use, and the MLIB-specific mechanisms offer finer control within that maximum. Refer to the *mpa*(1) or *f77*(1) man page, respectively, for details.

New Functionality

The sparse linear equation solver, described in Chapter 6 of the HP MLIB VECLIB User's Guide and the *sparse-solver*(3m) man page, has been parallelized for SPP1200 and SPP1600 systems, and several new subprograms have been added to give the user additional control over the package or additional output from it. These are documented in the following new man pages: *dsleff*(3m), *dslemp*(3m), *dsleop*(3m), and *dslerd*(3m). This functionality was provided in the HP MLIB V4.0 release for S-Class systems.

This HP MLIB release includes subprograms that compute tapered convolutions. These are documented in the *stconv*(3m) and *dtconv*(3m) man pages.

CPUTIME and WALLTIME are utility subprograms included in the VECLIB and LAPACK libraries. They make it easy to measure intervals of CPU time and wallclock time, respectively, in a program. See the *cputime*(3m) and *walltime*(3m) man pages for details.

HP MLIB V4.1 includes the SCILIB library. SCILIB is a collection of Fortran-callable mathematical subprograms which provides a look-alike implementation of the Scientific Library portion of Cray Research's UNICOS

Math and Scientific Library, V5.0. SCILIB had been provided as part of ConvexMLIB V3.0 for SPP1200 and SPP1600 systems, but is new to S- and X-Class systems.

Parallelized Subprograms

Tables 1-3 list the name, chapter of the user's guide, and purpose of various subroutines in LAPACK, SCILIB, and VECLIB that have been parallelized to improve performance on Exemplar servers. Many additional subprograms in the libraries call the BLAS, and will inherit their parallelism.

Table 1: Parallelized Subprograms in LAPACK

Subroutine name	Chapter	Description
{S,D,C,Z}GBSV	2	Solve a General Band Linear System
{S,D,C,Z}GESV	2	Solve a General Full Linear System
{S,D,C,Z}GBSVX	3	Solve a General Band Linear System
{S,D,C,Z}GESVX	3	Solve a General Full Linear System
S,D,C,Z}GBTRF	4	Factor a General Band Matrix
{S,D,C,Z}GETRF	4	Factor a General Full Matrix
{S,D,C,Z}GETRI	4	Invert General Matrix
{S,D,C,Z}GETRS	4	Solve a General Full Linear System
{S,D}GELS	5	Solve General Least Squares Problem
{S,D}GELSS	5	Solve General Least Squares Problem
{S,D}GEQRF	6	QR Factorization of a General Matrix
{S,D}GGQRF	6	Generalized QR Factorization
{S,D}GGRQF	6	Generalized RQ Factorization
{S,D}GEGS	9	Generalized Schur Form
{S,D}GEGV	9	Generalized General Matrix Eigenproblem
{S,D}GESVD	10	Singular Value Decomposition
{S,D,C,Z}GEMM	-	General Matrix-Matrix Multiply
{S,D,C,Z}GEMV	-	General Matrix-Vector Multiply
{S,D}GER	-	General Rank-1 Update
{C,Z}GERC	-	General Rank-1 Update
{C,Z}GERC	-	General Rank-1 Update
{C,Z}GERU	-	General Rank-1 Update

Table 2: Parallelized Subprograms in SCILIB

Subroutine name	Chapter	Description
CFFT2	6	Complex-to-Complex One-Dimensional FFT
CFFTMLT	6	Simultaneous Complex-to-Complex One-Dimensional FFT
CRFFT2	6	Complex-to-Real One-Dimensional FFT
RCFFT2	6	Real-to-Complex One-Dimensional FFT
RFFTMLT	6	Simultaneous Real-to-Complex One-Dimensional FFT

Table 3: Parallelized Subprograms in VECLIB

Subroutine name	Chapter	Description
{S,D,C,Z}GEMM	3	General Matrix-Matrix Multiply
{S,D,C,Z}GEMV	3	General Matrix-Vector Multiply
{S,D}GER	3	General Rank-1 Update
{C,Z}GERC	3	General Rank-1 Update
{C,Z}GERU	3	General Rank-1 Update
{S,D,C,Z}GBFA	4	Factor a General Band Matrix
{S,D}GEFA	4	Factor a General Matrix
{S,D}GESL	4	Solve General Linear Equations
DSLEFA	6	Numeric Factor Sparse Matrix
DSLEFS	6	Solve Sparse Linear Equations
DSKYFA	8	Numeric Factor Skyline Matrix
DSKYFS	8	Solve Skyline Linear Equations
DSKYFX	8	Solve Skyline Linear Equations
{S,D,C,Z}1DFFT	9	One-dimensional FFT
{S,D,C,Z}2DFFT	9	Two-dimensional FFT
{S,D,C,Z}3DFFT	9	Three-dimensional FFT
{S,D,C,Z}FFTS	9d	Simultaneous Complex-to-Complex One-Dimensional FFT
{S,D,C,Z}R12FT	9	Real-to-Complex One-Dimensional FFT
{S,D,C,Z}RC2FT	9	Real-to-Complex Two-Dimensional FFT
{S,D,C,Z}RC3FT	9	Real-to-Complex Three-Dimensional FFT
{S,D,C,Z}RCFTS	9	Simultaneous Real-to-Complex One-Dimensional FFT
{S,D}CONV	10	Correlation/Convolution

Known Documentation Problems

As HP MLIB V4.1 is accompanied with the same user's guides as V4.0, the description of parallel processing at run time is incorrect in all three printed manuals. Please refer to "Controlling Parallelism at Run Time" in this Release Note or one of the man pages *lapack*(3m), *scilib*(3m), or *veclib*(3m) to learn how you can control HP MLIB parallelism at run time.

Contrary to what is written about profiling MLIB applications in Chapter 1 of each of the three HP MLIB User's Guides, MLIB libraries are not instrumented for profiling with CXpa. CXpa is an optional product. If you have it, you can use CXoi, the Exemplar PA-RISC Object and Archive File Instrumentor, to make instrumented MLIB libraries. See *CXoi*(1) for details.