digital™

Software Product Description

PRODUCT NAME: Digital Extended Math Library Version 3.3 for Digital UNIX **SPD 41.86.06**

DESCRIPTION

Digital Extended Math Library (DXML) is a set of mathematical subprograms that are optimized for Digital architectures. Included subprograms cover the areas of Basic Linear Algebra, Linear System and Eigenproblem Solvers, Sparse Linear System Solvers, Sorting, Random Number Generation, and Signal Processing.

The Basic Linear Algebra library includes the industry-standard Basic Linear Algebra Subprograms (BLAS) Level 1, Level 2, and Level 3. Also included are subprograms for BLAS Level 1 Extensions, Sparse BLAS Level 1, and Array Math Functions (VLIB).

The Linear System and Eigenproblem Solver library provides the complete LAPACK package developed by a consortium of university and government laboratories. LAPACK is a new, industry-standard subprogram package offering an extensive set of linear system and eigenproblem solvers. LAPACK uses blocked algorithms that are better suited to most modern architectures, particularly ones with memory hierarchies. LAPACK will supersede LINPACK and EISPACK for most users.

The Sparse Linear System library provides both direct and iterative sparse linear system solvers. The direct solver package supports both symmetric and nonsymmetric sparse matrices stored using the skyline storage scheme. The iterative solver package contains a basic set of storage schemes, preconditioners, and iterative solvers. The design of this package is modular and matrix-free, allowing future expansion and easy modification by users.

The Signal Processing library provides a basic set of signal processing functions. Included are one-, two-, and three-dimensional Fast Fourier Transforms (FFT), group FFTs, Cosine/Sine Transforms (FCT/FST), Convolution, Correlation, and Digital Filters.

Many DXML subprograms are optimized for the supported hardware platforms. Optimization techniques include traditional optimizations such as loop unrolling and loop reordering. DXML subprograms also provide efficient management of the hierarchical memory system, using techniques such as the following:

- Reuse of data within registers to minimize memory accesses
- Efficient cache management
- Use of blocked algorithms that minimize translation buffer misses and unnecessary paging

Since DXML routines can be called from all languages that support the Digital UNIX® calling standard, the library provides optimized computation for applications written in these languages. Where appropriate, most subprograms are available in both real and complex versions, as well as in both single and double precision. The supported floating point format is IEEE.

Parallel Library Support for Symmetric Multiprocessing

DXML also supports symmetric multiprocessing (SMP) for improved performance. Key BLAS Level 2 and 3 routines, the LAPACK GETRF and POTRF routines, the sparse iterative solvers, the skyline solvers, and the FFT routines have been modified to execute in parallel if run on SMP hardware. These parallel routines along with the other serial routines are supplied in an alternative library. The user may choose to link with either the parallel or the serial library, depending on whether SMP support is required, since each library contains the complete set of routines.

DXML Run-Time Only Option

Digital provides a DXML Run-Time Only Option to allow applications built against the DXML shared library to be run on other systems. Each additional target system must have a DXML Run-Time Only Option (library and license) installed in order to run applications built with the Development Option. The DXML Run-Time Only Option does not permit new applications to be developed.

Distributing Applications Built with the DXML Run-Time Library

The Digital Extended Math Library is an application development tool that provides convenience and improved performance to the developer. To encourage application developers to incorporate DXML routines from the DXML archive libraries into their applications for distribution to other users, Digital permits the distribution of the DXML Run-Time Library (RTL), under the following conditions.

You may copy and distribute royalty-free the DXML RTL provided that you:

1. distribute the RTL only in conjunction with and as a part of your application,
2. include Digital's copyright notice on each copy of your application,
3. do not use Digital's logo or trademarks to market your application, and
4. agree to defend and indemnify Digital from and against any claims or lawsuits that arise or result from the use or distribution of your application.

The Run-time Library is that portion of the DXML Software that is required during the execution of your application. For V3.3, the RTL components are defined to be:

- libdxml_ev4.a
- libdxml_ev5.a

Basic Linear Algebra Subprograms

Linear algebra operations are fundamental to many mathematical applications, and several libraries of linear algebra subprogramss exist throughout the computer industry. The DXML BLAS library contains the most commonly used linear algebra subprograms.

The DXML linear algebra library contains five groups of subprograms at three levels:

- Basic Linear Algebra Subprograms (BLAS) Level 1
- BLAS Level 1 Extensions
- BLAS Level 1 Sparse Extensions
- BLAS Level 2
- BLAS Level 3

BLAS Level 1 (Scalar/Vector and Vector/Vector Operations)

BLAS Level 1 provides a set of elementary vector functions, operating on one or two vectors. These are typically very small routines, and they make less efficient use of the computing resources of modern computer architectures than the Level 2 and 3 operations.

DXML provides the 15 standard BLAS Level 1 operations:

- The index of the element of a vector having maximum absolute value
- The sum of the absolute values of the elements of a vector
- Inner product of two real vectors
- Scalar plus the extended precision inner product of two real vectors
- Conjugated inner product of two complex vectors
- Unconjugated inner product of two complex vectors
- Square root of the sum of squares (norm) of the elements of a vector
- Scalar times a vector plus a vector
- Copy one vector to another
- Apply a Givens rotation
- Apply a modified Givens plane rotation
- Generate elements for a Givens plane rotation
- Generate elements for a modified Givens plane rotation
- Product of a vector times a scalar
- Swap the elements of two vectors

BLAS Level 1 Extensions (Vector/Vector Operations)

When developing mathematical algorithms using the BLAS Level 1, scientists and engineers found that several additional constructs were used on a regular basis. These constructs are well known throughout the computer industry as BLAS Level 1 Extensions.

DXML contains 13 BLAS Level 1 Extension operations:

- Index of element having the minimum absolute value
- Index of element having the maximum value
- Index of element having the minimum value
- Largest value of the elements of a vector
- Smallest value of the elements of a vector
- Largest absolute value of the elements of a vector
- Smallest absolute value of the elements of a vector
- Sum of the values of the elements of a vector
- Set all elements of a vector equal to a scalar
- Constant times a vector set to another vector ($y = a*x$)
- Euclidean norm with no intermediate scaling
- Sum of the squares of the elements of a vector

- Constant times a vector plus a vector set to another vector (z = a*x + y)

BLAS Level 1 Sparse Extensions (Vector/Vector Operations)

This group of operations is similar to the BLAS Level 1 routines, but is designed to work on sparse vectors (vectors in which most of the elements are zero). Six of the routines are from industry standard Sparse BLAS 1, and the remaining three are enhancements.

The nine sparse BLAS Level 1 operations are:

- Scalar times a sparse vector plus a vector
- Sum of a sparse vector and a full vector
- Inner product of a sparse vector and a full vector
- Gather a sparse vector from a full vector
- Gather a sparse vector from the scaled elements of a full vector
- Gather a sparse vector from a full vector and zero corresponding elements of full vector
- Apply Givens rotation to a sparse vector and a full vector
- Scatter a sparse vector into a full vector
- Scale and scatter a sparse vector into a full vector

BLAS Level 2 (Matrix/Vector Operations)

The BLAS Level 2 codes make more effective use of the data in the registers, reducing the number of register loads and stores required. In addition, loop unrolling techniques are used to minimize cache misses and page faults. The BLAS Level 2 subprograms use the following types of operations:

- Matrix/vector products
- Rank-1 and rank-2 matrix updates
- Solutions of triangular systems of equations

Six types of matrices are supported by these BLAS Level 2 routines:

- General
- General band
- Symmetric/Hermitian
- Symmetric/Hermitian band
- Triangular
- Triangular band

BLAS Level 3 (Matrix/Matrix Operations)

The BLAS Level 3 routines operate at a level that makes the most efficient use of machine resources. DXML optimizes these routines by partitioning matrices into blocks and computing matrix/matrix operations on each block. This approach avoids excessive memory accesses by providing full reuse of data while each block is in the cache or the registers. BLAS Level 3 routines provide this kind of blocking for three basic types of operations:

- Matrix/matrix products
- Rank-k and rank-2k updates of a symmetric matrix
- Solving triangular systems of equations with multiple right-hand sides

Three types of matrices are supported by these BLAS Level 3 routines:

- General
- Symmetric/Hermitian
- Triangular

A set of additional matrix-matrix routines is provided:

- Add two matrices
- Subtract one matrix from another
- Transpose a matrix, in-place or out-of-place

Array Math Functions

The Array Math Functions provide a set of basic math functions that operate on arrays of numbers rather than on scalars. On vector and superscalar architectures, such functions have a performance advantage over a loop of scalar operations. The library includes the following array functions for double precision numbers:

- Sine of array
- Cosine of array
- Cosine and sine of array
- Exponent of array
- Logarithm of array
- Square root of array
- Reciprocal of array

LAPACK Library Contents

LAPACK is a library of linear algebra subprograms intended to solve a wide range of problems in linear algebra. LAPACK can be used to solve dense systems of linear equations, linear least squares problems, eigenvalue problems, and singular value problems. It is also useful in doing other computations such as matrix factorizations and estimations of condition numbers.

The DXML LAPACK library provides the complete LAPACK package. DXML's version of LAPACK is provided as a packaged library, compiled, tested, and ready-to-use. Combined with the optimized BLAS Level 3 routines, the DXML LAPACK will provide optimal performance on all supported platforms. LAPACK should be used in place of LINPACK and EISPACK, because it is more efficient, accurate, and robust.

LAPACK supports both real and complex, single and double precision data. It operates on the following types of matrices:

- Bidiagonal
- General band
- General unsymmetric
- General tridiagonal
- Hermitian
- Hermitian, packed storage
- Upper Hessenberg, generalized problem
- Upper Hessenberg
- Orthogonal
- Orthogonal, packed storage
- Symmetric/Hermitian positive definite band
- Symmetric/Hermitian positive definite
- Symmetric/Hermitian positive definite, packed storage
- Symmetric/Hermitian positive definite tridiagonal
- Symmetric band
- Symmetric, packed storage
- Symmetric tridiagonal
- Symmetric
- Triangular band
- Triangular, generalized problem
- Triangular, packed storage
- Triangular
- Trapezoidal
- Unitary
- Unitary, packed storage

LAPACK provides the following operations:

- Triangular factorization
- Unblocked triangular factorization
- Solve a system of linear equations (based on triangular factorization)
- Compute the inverse (based on triangular factorization)
- Compute a split Cholesky factorization of a symmetric/Hermitian positive definite band matrix
- Unblocked computation of inverse
- Estimate condition number
- Refine initial solution returned by solver
- Perform QR factorization without pivoting
- Unblocked QR factorization
- Solve linear least squares problem (based on QR factorization)
- Solve the linear equality constrained least squares (LSE) problem
- Solve the Gauss-Markov linear model problem
- Perform LQ factorization without pivoting
- Unblocked LQ factorization
- Solve underdetermined linear system (based on LQ factorization)
- Generate a real orthogonal or complex unitary matrix as a product of Householder matrices
- Unblocked generation of real orthogonal or unitary matrix
- Multiply a matrix by a real orthogonal or complex unitary matrix by applying a product of Householder matrices
- Unblocked version of multiplication of a matrix by a real orthogonal or complex unitary matrix by applying a product of Householder matrices
- Reduce a square matrix to upper Hessenberg form
- Unblocked version of square matrix reduction
- Reduce a symmetric matrix to real symmetric tridiagonal form
- Reduce a band matrix to bidiagonal form
- Unblocked version of symmetric matrix reduction
- Reduce a rectangular matrix to bidiagonal form
- Reduce a band symmetric/Hermitian matrix to tridiagonal form
- Reduce a symmetric/Hermitian-definite banded generalized eigenproblem to standard form
- Compute various norms of a complex Hermitian tridiagonal matrix
- Compute eigenvalues and optional Schur factorization or eigenvectors using QR algorithm
- Compute selected eigenvectors by inverse iteration
- Compute eigenvectors from Schur factorization

- Compute eigenvectors using the Pal-Walker-Kahan variant of the QL or QR algorithm
- For a pair of N-by-N real nonsymmetric matrices, compute the generalized eigenvalues, the real Schur form, and the left and/or right Schur vectors
- For a pair of N-by-N real nonsymmetric matrices, compute the generalized eigenvalues, and the left and/or right generalized eigenvectors
- Solve the generalized nonsymmetric eigenproblem Ax = lambda Bx
- Solve the generalized definite banded eigenproblem Ax = lambda Bx
- Solve the generalized symmetric/Hermitian-definite banded eigenproblem
- Solve the symmetric eigenproblem using divide-and-conquer algorithm
- Compute singular values and, optionally, singular vectors using the QR algorithm
- Compute the generalized (quotient) singular value decomposition
- Compute the generalized singular value decomposition (GSVD) on the M-by-N matrix A and P-by-N matrix B
- Solve a generalized linear regression model problem

Sparse System Solver Subrograms

The DXML Sparse System Solver library contains a set of subprograms that may be used to solve sparse linear systems of equations. Two packages providing direct and iterative methods are supported.

Direct Method Sparse Solver Package

The direct solver package includes skyline (profile) solvers for symmetric and nonsymmetric matrices. Separate factorization and solver routines are provided to allow repeated use of the solver for multiple right hand sides, without repeating the factorization. To make the subprograms easier to use, both simple and expert driver routines are provided. Functions provided include:

- LDU factorization
- Solve
- Norm evaluation
- Condition number estimation
- Iterative refinement
- Simple and expert drivers

These storage schemes are supported for symmetric and nonsymmetric matrices:

- Profile-in storage
- Structurally symmetric, profile-in storage (for nonsymmetric only)
- Diagonal-out storage

Iterative Method Sparse Solver Package

For the iterative method, the library provides a modular set of storage schemes, preconditioners, and solvers. These solvers and preconditioners are easily accessed through an integrated driver routine.

Six iterative sparse solvers for real, double precision data are supplied:

- Preconditioned conjugate gradient method
- Preconditioned least squares conjugate gradient method
- Preconditioned biconjugate method
- Preconditioned conjugate gradient squared method
- Preconditioned generalized minimum residual method
- Preconditioned transpose free QMR method

Routines for three storage schemes are provided, or the user may develop routines to employ a custom storage scheme. The supplied storage schemes include:

- Symmetric diagonal
- Unsymmetric diagonal
- General storage by rows

Three preconditioners are supplied, which can be selectively applied to the data. Users may also supply custom preconditioners. The preconditioners supplied include:

- Diagonal
- Polynomial (Neumann)
- Incomplete LU with zero diagonals added

Sorting Subprograms

Two sort subprograms using the Quicksort algorithm and two general purpose radix sort subprograms are provided, as follows:

- Sort elements of a vector using the Quicksort algorithm
- Sort an indexed vector of data using the Quicksort algorithm
- Sort data using a radix sort algorithm

- Sort an indexed vector of data using a radix sort algorithm

All of the above sorts operate on data stored in memory.

Random Number Subprograms

DXML provides four random number generator subprograms:

- Produce a vector of uniform [0,1], long-period random numbers using the L'Ecuyer multiplicative method
- Produce a vector of N(0,1), normally-distributed random numbers

Note: Two auxilliary input routines are provided to allow the above generator subprograms to be called from within a parallel section of a program.

- Produce single precision random numbers using a linear multiplicative algorithm
- Produce single precision random numbers using a Lehmer multiplicative generator

Signal Processing Subprograms

The DXML Signal Processing library contains a set of subprograms in four basic areas of signal processing:

- Fast Fourier Transforms (FFT)
- Fast Cosine and Fast Sine Transforms (FCT and FST)
- Convolution and correlation
- Digital filters

Fast Fourier Transforms and Cosine and Sine Transforms

DXML provides one-dimensional, two-dimensional, three-dimensional, and group FFT routines and one-dimensional FCT/FST routines. Each routine is supplied in two forms:

- The first form computes the transform in one unit operation. This is convenient for programs requiring speed on only one or a few operations.
- The second form is provided for programs requiring speed on repeated operations. With this form, each routine is subdivided into three routines. One routine builds the rotation factors, a second routine applies them to perform the transform, and a third routine deallocates any virtual memory allocated in the first routine. Thus, for repeated operations, the rotation factors need to be built only once.

Convolution and Correlation

DXML provides routines for computing one-dimensional discrete convolutions and correlations. These routines can process both periodic and nonperiodic data.

Digital Filters

DXML provides support for one-dimensional, nonrecursive digital filtering. Based on the Kaisers Sinh-Bessel algorithm, these routines allow programming of bandpass, bandstop, low-pass, and high-pass filters.

Cray LibSci Portability Support

SCIPORT is a Digital Equipment Corporation implementation of the Cray Research scientific numerical library, LibSci. SCIPORT provides 64-bit, single-precision library routines for Cray users porting programs to Alpha systems running Digital UNIX. SCIPORT also provides equivalent versions of almost all Cray Math Library and CF77 (Cray Fortran 77) Math intrinsic routines. SCIPORT is provided as an optional subset of DXML.

SCIPORT provides the following:

- True 64-bit versions of all Cray LibSci single-precision BLAS Level 1, Level 2, and Level 3 routines
- All Cray LibSci LAPACK routines
- All Cray LibSci Special Linear System Solver routines
- All Cray LibSci Signal Processing routines
- All Cray LibSci Sorting and Searching routines

These routines are completely interchangeable with their Cray LibSci counterparts and, with the exception of the ORDERS routine, require no program changes to function correctly. Owing to endian differences of machine architecture, special considerations must be given when the ORDERS routine is used to sort multi-byte character strings.

HARDWARE REQUIREMENTS

DXML will operate on any AlphaStation or AlphaServer capable of running Digital UNIX. In addition, DXML will operate correctly when the archive library is linked to an application built with the Digital UNIX version of the VxWorks® development environment and executed on an Alpha embedded processor. Such use may require an additional license.

DXML versions 3.1-3.3 provide two versions of the libraries built for the Alpha EV4 and EV5 implementations. Both versions of the libraries will function correctly on either EV4 or EV5 processors, but may exhibit some performance loss when not run on the designated implementation.

Disk Space Requirements

Development Option

Disk space required for installation:

Root file system:	/ 0 MB
Other file systems:	/usr 90 MB
	/tmp 0 MB
	/var 0 MB

Disk space required for use (permanent), including man pages:

Root file system:	/ 0 MB
Other file systems:	/usr 57 MB
	/var 0 MB

Run-Time Option

Disk space required for installation:

Root file system:	/ 0 MB
Other file systems:	/usr 55 MB
	/tmp 0 MB
	/var 0 MB

Disk space required for use (permanent):

Root file system:	/ 0 MB
Other file systems:	/usr 20 MB
	/var 0 MB

These counts refer to the disk space required on the system disk. The sizes are approximate; actual sizes may vary depending on the user's system environment, configuration, and software options.

SOFTWARE REQUIREMENTS

Digital UNIX Operating System Version V3.2-V3.2G or V4.0-V4.0A

GROWTH CONSIDERATIONS

The minimum hardware/software requirements for any future version of this product may be different from the requirements for the current version.

DISTRIBUTION MEDIA

This product is available as part of the Digital UNIX Consolidated Software Distribution on CD-ROM (QA-054AA-H8).

ORDERING INFORMATION

The software documentation for this product is also available as part of the Digital UNIX Online Documentation Library on CD-ROM.

Development Option

Software Licenses: QL-MUXA*-**

Software Media: QA-MUXAA-H8

Software Documentation: QA-MUXAA-GZ

Software Product Services: QT-MUXA*-**

Run-Time Option

Software Licenses: QL-MUYA*-**

Software Media: QA-MUYAA-H8

Software Documentation: QA-MUYAA-GZ

Software Product Services: QT-MUYA*-**

* Denotes variant fields. For additional information on available licenses, services, and media, refer to the appropriate price book.

SOFTWARE LICENSING

This software is furnished only under a license. For more information about Digital's licensing terms and policies, contact your local Digital office.

License Management Facility Support

This layered product supports the Digital UNIX License Management Facility. License units for this product are allocated on an Unlimited Use Basis.

For more information on the License Management Facility, refer to the Digital UNIX Operating System Software Product Description (SPD 41.61.xx) or to the Digital UNIX Operating System documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available. For more information, please contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum.

The above information is valid at time of release. Please contact your local Digital office for the most up-to-date information.