

## TechReview, Part 1: Discover the LAPACK library

Learn about the four versions of this interface for commonly used linear algebra operations in high performance computing

Skill Level: Introductory

[Kane Scarlett \(kane@us.ibm.com\)](mailto:kane@us.ibm.com)

Editor, Multicore acceleration

IBM

02 Sep 2008

For application programmers using the IBM Software Development Kit for Multicore Acceleration (SDK), this article explains the basic structure of the IBM Linear Algebra Package (LAPACK) library. The LAPACK is based on a published standard interface for commonly used linear algebra operations in high performance computing and other scientific domains.

### Introduction

This review gives a quick look at the LAPACK library through the eyes of the original documentation, "LAPACK: Linear Algebra Package Library Programmer's Guide and API Reference" (see [Resources](#)). The article focuses on introducing the library versions and functions and on overviewing the basic structure of the library. Use this article with the most current version of IBM SDK for Multicore Acceleration (also known as the Cell/B.E.® SDK), which is the version with fixpack 3.0.0.3.

### Discovering LAPACK

LAPACK is a software library that provides routines for solving systems of:

- Simultaneous linear equations: Algebraic equations in which each term is a constant (or the product of a constant) and a single variable (with one or

more variables).

- Least-squares solutions of linear systems of equations: A method of fitting data often used in statistical contexts such as regression analysis.
- Eigenvalue problems: The corresponding scalar value of the *eigenvector*, which is a nonzero vector that changes in length (but not direction) when applied to the linear transformation of which it is a part.
- Householder transformation: The reflection of a vector in a plane in 3D space. Householder transformation is often used to implement decomposition on a matrix.

LAPACK is written in FORTRAN 77. LAPACK95 uses features of FORTRAN 95 to simplify the interface of the routines. In a way, LAPACK is the successor of LINPACK, which was designed to run on vector computers with shared memory. One of the major differences, though, is that LAPACK depends on Basic Linear Algebra Subprograms (see [Resources](#) for more information about BLAS) to leverage the cache found in modern cache-based systems architectures. (And, if your BLAS is well tuned, LAPACK leaves LINPACK in the dust!) The library can also run on distributed-memory systems, including ScaLAPACK and PLAPACK.

The API is available with standard ANSI C and standard FORTRAN 77 interfaces. Implementations of the APIs are available as open source from Netlib Repository.

## Versions

Each LAPACK routine has up to four versions:

- *Real single precision* is denoted by the prefix *S*.
- *Real double precision* is denoted by the prefix *D*.
- *Complex single precision* is denoted by the prefix *C*.
- *Complex double precision* is denoted by the prefix *Z*.

The LAPACK library in the Cell/B.E. SDK supports only real double precision (or DP). DP routines are available as PPE APIs, and the routines conform to the standard LAPACK FORTRAN 77 interface.

## Optimized routines

The following routines have been optimized to use features of the Synergistic Processing Elements (SPEs):

- `DGETRF` : Compute the LU factorization of a general matrix. LU is a matrix decomposition that writes a matrix as the product of a lower and upper

triangular matrix.

- **DGETRI**: Compute the inverse of a general matrix using the LU factorization.
- **DGEQRF**: Compute the QR factorization of a general matrix. QR is a decomposition of the matrix into an orthogonal and a triangular matrix.
- **DPOTRF**: Compute the Cholesky factorization of a symmetric positive matrix. Cholesky is a decomposition of a symmetric positive-definite matrix into a lower triangular matrix and the transpose of the lower triangular matrix.
- **DBDSQR**: Compute the singular value decomposition of a real bi-diagonal matrix using the implicit zero-shift QR algorithm.
- **DSTEQR**: Compute the singular value decomposition of a real symmetric tridiagonal matrix using the implicit QR algorithm.

## Understanding the basic library structure

Table 1 shows where to find various library, header, and code example files in the LAPACK library for Cell/B.E. and x86 systems.

**Table 1. Where the files are**

Platform	Cell/B.E. or Power host (development or execution, including simulator)	x86 or x86_64 (development)
<b>PPE 32-bit library</b>	/usr/lib/liblapack.a /usr/lib/liblapack.so /usr/lib/liblapack.so.3.1.1	/opt/cell/sysroot/usr/lib/liblapack.a /opt/cell/sysroot/usr/lib/liblapack.so /opt/cell/sysroot/usr/lib/liblapack.so.3.1.1
<b>PPE 64-bit library</b>	/usr/lib64/liblapack.a /usr/lib64/liblapack.so /usr/lib64/liblapack.so.3.1.1	/opt/cell/sysroot/usr/lib64/liblapack.a /opt/cell/sysroot/usr/lib64/liblapack.so /opt/cell/sysroot/usr/lib64/liblapack.so.3.1.1
<b>PPE header files</b>	/usr/include/lapack.h	/opt/cell/sysroot/usr/include/lapack.h
<b>Code examples</b>	/opt/cell/sdk/src/lapack-examples-source.tar	

The following are the key file components of the LAPACK library:

- lapack.h contains the C function interface of LAPACK on PPE for DP.
- lapack.a contains the static library, which contains the LAPACK library for Cell/B.E.
- lapack.so is a shared LAPACK library for Cell/B.E.

- lapack-examples-source.tar contains two examples that show how to use the LAPACK library with the SDK for Multicore Acceleration.

## Conclusion

The next TechReview of LAPACK introduces some basic programming and performance tuning tips for the library. If the anticipation of it grows unbearable, you can jet over to the original source document (see [Resources](#)).

# Resources

## Learn

- Use an [RSS feed](#) to request notification for the upcoming articles in this series. (Find out more about [RSS feeds of developerWorks content](#).)
- Get the source document for this article, "[LAPACK: Linear Algebra Package Library Programmer's Guide and API Reference](#)" (IBM, April 2008), to learn how to configure the LAPACK library and how to program applications that use LAPACK on the IBM SDK for Multicore Acceleration, Version 3.0.0.3. The guide contains reference information about APIs for the library, and it contains sample applications showing usage of these APIs.
- See [Programming with BLAS: The series](#) (developerWorks, November 2007-July 2008) for a roundup of short guides and longer articles to help you understand and use BLAS. (There are two other tech topic roundups available in this format too: [Programming with ALF](#) and [Programming with DaCS](#).)
- Learn more about Cell/B.E. programming from the developerWorks series:
  - "[Programming high-performance applications on the Cell/B.E. processor](#)"
  - "[PS3 fab-to-lab](#)"
  - "[The little broadband engine that could](#)"
- Refer to the [Cell Broadband Engine documentation](#) section of the IBM Semiconductor Solutions Technical Library for a wealth of downloadable manuals, specifications, and more.
- Sign up for the [developerWorks newsletter](#) and get the latest developer news and Cell/B.E. happenings delivered to your inbox each week. Check *Power Architecture*® when you sign up to receive Cell/B.E. news in your newsletter.

## Get products and technologies

- Find the [LAPACK routine](#) you need at the [LAPACK section of the Netlib Repository](#), as well as a [user's guide](#), a [user forum](#), a [quick installation guide](#), and more.
- Look for [ScaLAPACK](#) (or Scalable LAPACK) for a subset of LAPACK routines redesigned for distributed-memory MIMD parallel computers.
- Discover [PLAPACK](#) (the Parallel Linear Algebra Package) for an infrastructure for coding linear algebra algorithms at a high level of abstraction.
- Get your copy of the [IBM SDK for Multicore Acceleration 3.0](#) or browse through the [library of Cell/B.E. documentation](#).
- Find all Cell/B.E.-related articles, discussion forums, downloads, and more at

the IBM developerWorks [Cell Broadband Engine resource center](#): your definitive resource for all things Cell/B.E.

- Contact IBM about [custom Cell/B.E.-based or custom-processor based solutions](#).

## Discuss

- Check out the [Cell Broadband Engine Architecture forum](#) to get your technical questions about the processor answered. Juicy problems and answers from the forums are rounded up periodically and highlighted in the "[Forum watch](#)" [blog series](#).
- Go to the [Cell Broadband Engine/Power Architecture blog](#) for [news](#), downloads, instructional resources, and event notifications for Cell/B.E. and other Power Architecture-related technologies. You can find the popular "[Forum watch](#)" [blog series](#) (Q&A roundup), the "[FixIt](#)" technology updates, and the [Infobomb](#) quick-read technology introductions.

## About the author

Kane Scarlett

Kane Scarlett is a technology journalist/analyst with 20 years in the business, working for such publishers as National Geographic, Population Reference Bureau, Miller Freeman, and IDG, and managing, editing, and writing for such august journals as *JavaWorld*, *LinuxWorld*, and of course, developerWorks.

## Trademarks

IBM, developerWorks, and Power Architecture are trademarks of IBM Corporation in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc.

Other company, product, or service names may be trademarks or service marks of others.