Title

> **LAPACK —** Linear algebra package (LAPACK) routines

Description

LAPACK stands for Linear Algebra PACKage and is a freely available set of Fortran 90 routines for solving systems of simultaneous equations, eigenvalue problems, and singular value problems. Many of the LAPACK routines are based on older EISPACK and LINPACK routines, and the more modern LAPACK does much of its computation by using Basic Linear Algebra Subprograms (BLAS).

Stata contains two sets of LAPACK and BLAS libraries; one is from Netlib, and the other is from the Intel Math Kernel Library (MKL).

set lapack_mkl sets which LAPACK library will be used.

set lapack_mkl_cnr sets the conditional numerical reproducibility mode for the Intel MKL LAPACK routines.

Syntax

Set whether Intel MKL LAPACK routines will be used

set lapack_mkl { on | off } [, permanently]

Set the conditional numerical reproducibility mode for the Intel MKL LAPACK routines

set lapack_mkl_cnr { default | auto | compatible | off }

Option for set lapack_mkl

permanently specifies that, in addition to making the change right now, the setting be remembered and become the default setting when you invoke Stata in the future.

Remarks and examples

Remarks are presented under the following headings:

> *LAPACK in Mata*
> *set lapack_mkl*
> *Intel MKL conditional numerical reproducibility*
> *set lapack_mkl_cnr*

LAPACK in Mata

The LAPACK and BLAS routines form the basis for many of Mata's linear algebra capabilities. Individual functions of Mata that use LAPACK routines always make note of that fact.

Stata on all platforms, except ARM-based Mac (Apple Silicon), contains two sets of LAPACK and BLAS libraries. One is based on the source code from Netlib's LAPACK and has been used in Stata since Stata 9. Stata, since Stata 17, also contains LAPACK and BLAS libraries from the Intel MKL. Because the Intel MKL does not support ARM-based Mac, ARM-based Mac supports only Netlib's LAPACK library.

For up-to-date information on LAPACK, see http://www.netlib.org/lapack/.

For up-to-date information on Intel MKL, see https://software.intel.com/mkl.

Advanced programmers can directly access the LAPACK functions; see [M-5] **lapack( )**.

set lapack_mkl

For platforms that support both LAPACK libraries, the default is `set lapack_mkl on`, meaning the Intel MKL LAPACK routines are used. To instead use Netlib's LAPACK library, you can `set lapack_mkl off`. To determine which library is being used, check the contents of `c(lapack_mkl)`. On ARM-based Mac, `lapack_mkl` is not settable and is always off.

Note that `set lapack_mkl` should be specified in Stata, not in Mata.

Intel MKL conditional numerical reproducibility

Regardless of which LAPACK routine you use, you may encounter slight numeric differences when you run the same code in the same version and platform of Stata on a different OS. This is normal and can be caused by several reasons. For example, Stata is compiled with different compilers on different operating systems, and different compilers could produce slightly different numeric results.

You might also get different results on the same operating system if you use the Intel MKL routines. Rosenquist (2011) explains why: "To get the best performance wherever a program is run, Intel MKL will check on the processor type at run time and can dispatch processor-specific code accordingly. If a particular instruction set or cache of a certain size is available a specialized code path may exploit it. These code paths are different enough that they will again cause a different order of operations and will cause slightly different results on different processors." This means that there could be slight numeric differences when running the same code on two machines with the same flavor of Stata, the same operating system, but with different CPUs. That said, you can obtain different levels of numerical reproducibility with `set lapack_mkl_cnr`, discussed below.

set lapack_mkl_cnr

Intel offers a couple of different conditional numerical reproducibility (CNR) functions for obtaining different levels of reproducibility, and the type of reproducibility desired can be specified with the `lapack_mkl_cnr` setting. The `lapack_mkl_cnr` setting should be specified in Stata, not Mata, and it will take effect the next time you launch Stata. The possible values are `default`, `auto`, `compatible`, or `off`.

`set lapack_mkl_cnr default` sets the reproducibility level to SSE4_2 on Intel CPUs. On non-Intel CPUs, `set lapack_mkl_cnr default` is the same as `set lapack_mkl_cnr auto`.

set lapack_mkl_cnr auto allows the Intel MKL to automatically determine the code path based on the different features the CPU supports. The setting takes advantage of the specific features of the CPU, hence the improved performance. The cost is the lower level of numerical reproducibility; the same LAPACK function may return different results on different CPUs, assuming you are using the function on the same operating system and flavor of Stata.

Note that the automatically determined code path will not necessarily be the most efficient. For example, we noticed that the AVX instruction set caused excessively high CPU usage on Intel CPUs when Stata was idle.

set lapack_mkl_cnr compatible provides a higher level of numerical reproducibility compared with set lapack_mkl_cnr auto. In this setting, the LAPACK functions try to produce the same results by using a set of the most common features supported by different CPUs. This setting is also the slowest in terms of performance.

set lapack_mkl_cnr off turns off the code that tries to maintain conditional numerical reproducibility, which results in greater performance. With this setting, you may get different results across different runs of the same LAPACK function on the same copy of Stata and on the same machine.

For a detailed introduction to the Intel MKL conditional numerical reproducibility, see Rosenquist (2011).

Acknowledgments

We thank the authors of LAPACK for their excellent work:

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen.

References

Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. 1999. *LAPACK Users' Guide*. 3rd ed. Philadelphia: Society for Industrial and Applied Mathematics.

Rosenquist, T. 2011. Getting reproducible results with Intel MKL. https://software.intel.com/content/www/us/en/develop/articles/getting-reproducible-results-with-intel-mkl.html.

Also see

[M-5] **lapack( )** — Linear algebra package (LAPACK) functions

[R] **Copyright LAPACK** — LAPACK copyright notification

[M-1] **Intro** — Introduction and advice