
hipSOLVER Documentation

Release 1.8.0

Advanced Micro Devices

Feb 09, 2023

CONTENTS

1	hipSOLVER User Guide	3
1.1	Introduction	3
1.1.1	Library overview	3
1.1.2	Currently implemented functionality	4
	LAPACK auxiliary functions	4
	LAPACK main functions	4
	LAPACK-like functions	6
1.1.3	Compatibility-only functions	6
	Partial eigensolver functions	6
	Partial SVD functions	7
1.2	Installation	7
1.2.1	Install pre-built packages	7
1.2.2	Build & install library using script (Ubuntu only)	7
1.2.3	Build & install library manually	8
1.2.4	Build library + tests + benchmarks + samples manually	8
	Dependencies (only necessary for hipSOLVER clients)	8
	Library and clients	8
1.3	Using hipSOLVER	9
1.3.1	Porting cuSOLVER applications to hipSOLVER	9
1.3.2	Some considerations when using the compatibility hipSOLVER API	9
	Different minimum array lengths	10
	Arguments not referenced by rocSOLVER	10
	Possible performance implications of the compatibility API	10
1.3.3	Some considerations when using the regular hipSOLVER API	11
	Different signatures and additional API methods	11
	Arguments not referenced by rocSOLVER	12
	Using rocSOLVER's memory model	12
	Using rocSOLVER's in-place functions	12
2	hipSOLVER API	13
2.1	Types	13
2.1.1	hipSOLVER regular API types	13
	hipsolverHandle_t	14
	hipsolverSyejvInfo_t	14
	hipsolverStatus_t	14
	hipsolverOperation_t	14
	hipsolverFillMode_t	14
	hipsolverSideMode_t	15
	hipsolverEigMode_t	15
	hipsolverEigType_t	15

2.2	Helper Functions	15
2.2.1	Handle set-up and tear-down	15
	hipsolverCreate()	16
	hipsolverDestroy()	16
2.2.2	Stream manipulation	16
	hipsolverSetStream()	16
	hipsolverGetStream()	16
2.2.3	Gesvdj parameter manipulation	16
	hipsolverCreateGesvdjInfo()	17
	hipsolverDestroyGesvdjInfo()	17
	hipsolverXgesvdjSetMaxSweeps()	17
	hipsolverXgesvdjSetSortEig()	17
	hipsolverXgesvdjSetTolerance()	17
	hipsolverXgesvdjGetResidual()	17
	hipsolverXgesvdjGetSweeps()	17
2.2.4	Syevj parameter manipulation	17
	hipsolverCreateSyevjInfo()	18
	hipsolverDestroySyevjInfo()	18
	hipsolverXsyevjSetMaxSweeps()	18
	hipsolverXsyevjSetSortEig()	18
	hipsolverXsyevjSetTolerance()	18
	hipsolverXsyevjGetResidual()	18
	hipsolverXsyevjGetSweeps()	18
2.3	LAPACK Auxiliary Functions	18
2.3.1	Orthonormal matrices	18
	hipsolver<type>orgbr_bufferSize()	19
	hipsolver<type>orgbr()	19
	hipsolver<type>orgqr_bufferSize()	19
	hipsolver<type>orgqr()	19
	hipsolver<type>orgtr_bufferSize()	20
	hipsolver<type>orgtr()	20
	hipsolver<type>ormqr_bufferSize()	20
	hipsolver<type>ormqr()	20
	hipsolver<type>ormtr_bufferSize()	20
	hipsolver<type>ormtr()	21
2.3.2	Unitary matrices	21
	hipsolver<type>ungbr_bufferSize()	21
	hipsolver<type>ungbr()	22
	hipsolver<type>ungqr_bufferSize()	22
	hipsolver<type>ungqr()	22
	hipsolver<type>ungtr_bufferSize()	22
	hipsolver<type>ungtr()	22
	hipsolver<type>unmqr_bufferSize()	23
	hipsolver<type>unmqr()	23
	hipsolver<type>unmtr_bufferSize()	23
	hipsolver<type>unmtr()	23
2.4	LAPACK Functions	24
2.4.1	Triangular factorizations	24
	hipsolver<type>potrf_bufferSize()	24
	hipsolver<type>potrfBatched_bufferSize()	25
	hipsolver<type>potrf()	25
	hipsolver<type>potrfBatched()	25
	hipsolver<type>getrf_bufferSize()	26
	hipsolver<type>getrf()	26

	hipsolver<type>sytrf_bufferSize()	26
	hipsolver<type>sytrf()	26
2.4.2	Orthogonal factorizations	27
	hipsolver<type>geqrf_bufferSize()	27
	hipsolver<type>geqrf()	27
2.4.3	Problem and matrix reductions	27
	hipsolver<type>gebrd_bufferSize()	28
	hipsolver<type>gebrd()	28
	hipsolver<type>sytrd_bufferSize()	28
	hipsolver<type>hetrd_bufferSize()	28
	hipsolver<type>sytrd()	29
	hipsolver<type>hetrd()	29
2.4.4	Linear-systems solvers	29
	hipsolver<type>potri_bufferSize()	29
	hipsolver<type>potri()	30
	hipsolver<type>potrs_bufferSize()	30
	hipsolver<type>potrsBatched_bufferSize()	30
	hipsolver<type>potrs()	31
	hipsolver<type>potrsBatched()	31
	hipsolver<type>getrs_bufferSize()	31
	hipsolver<type>getrs()	32
	hipsolver<type><type>gesv_bufferSize()	32
	hipsolver<type><type>gesv()	32
2.4.5	Least-squares solvers	33
	hipsolver<type><type>gels_bufferSize()	33
	hipsolver<type><type>gels()	33
2.4.6	Symmetric eigensolvers	34
	hipsolver<type>syevd_bufferSize()	34
	hipsolver<type>heevd_bufferSize()	34
	hipsolver<type>syevd()	34
	hipsolver<type>heevd()	35
	hipsolver<type>sygvd_bufferSize()	35
	hipsolver<type>hegvd_bufferSize()	35
	hipsolver<type>sygvd()	35
	hipsolver<type>hegvd()	36
2.4.7	Singular value decomposition	36
	hipsolver<type>gesvd_bufferSize()	36
	hipsolver<type>gesvd()	36
2.5	LAPACK-like Functions	37
2.5.1	Symmetric eigensolvers	37
	hipsolver<type>syevj_bufferSize()	37
	hipsolver<type>heevj_bufferSize()	38
	hipsolver<type>syevjBatched_bufferSize()	38
	hipsolver<type>heevjBatched_bufferSize()	38
	hipsolver<type>syevj()	38
	hipsolver<type>heevj()	39
	hipsolver<type>syevjBatched()	39
	hipsolver<type>heevjBatched()	39
	hipsolver<type>sygvj_bufferSize()	39
	hipsolver<type>hegvj_bufferSize()	40
	hipsolver<type>sygvj()	40
	hipsolver<type>hegvj()	40
2.5.2	Singular value decomposition	40
	hipsolver<type>gesvdj_bufferSize()	41

	hipsolver<type>gesvdjBatched_bufferSize()	41
	hipsolver<type>gesvdj()	42
	hipsolver<type>gesvdjBatched()	42
3	hipSOLVER Compatibility API	43
3.1	Types	43
3.1.1	hipSOLVER compatibility API types	43
	hipsolverDnHandle_t	44
	hipsolverGesvdjInfo_t	44
	hipsolverSyevejInfo_t	44
	hipsolverStatus_t	44
	hipsolverOperation_t	44
	hipsolverFillMode_t	44
	hipsolverSideMode_t	44
	hipsolverEigMode_t	44
	hipsolverEigType_t	44
	hipsolverEigRange_t	44
3.2	Helper Functions	45
3.2.1	Handle set-up and tear-down	45
	hipsolverDnCreate()	45
	hipsolverDnDestroy()	45
3.2.2	Stream manipulation	45
	hipsolverDnSetStream()	45
	hipsolverDnGetStream()	46
3.2.3	Gesvdj parameter manipulation	46
	hipsolverDnCreateGesvdjInfo()	46
	hipsolverDnDestroyGesvdjInfo()	46
	hipsolverDnXgesvdjSetMaxSweeps()	46
	hipsolverDnXgesvdjSetSortEig()	46
	hipsolverDnXgesvdjSetTolerance()	46
	hipsolverDnXgesvdjGetResidual()	47
	hipsolverDnXgesvdjGetSweeps()	47
3.2.4	Syevej parameter manipulation	47
	hipsolverDnCreateSyevejInfo()	47
	hipsolverDnDestroySyevejInfo()	47
	hipsolverDnXsyevejSetMaxSweeps()	47
	hipsolverDnXsyevejSetSortEig()	47
	hipsolverDnXsyevejSetTolerance()	48
	hipsolverDnXsyevejGetResidual()	48
	hipsolverDnXsyevejGetSweeps()	48
3.3	LAPACK Auxiliary Functions	48
3.3.1	Orthonormal matrices	48
	hipsolverDn<type>orgbr_bufferSize()	49
	hipsolverDn<type>orgbr()	49
	hipsolverDn<type>orgqr_bufferSize()	49
	hipsolverDn<type>orgqr()	49
	hipsolverDn<type>orgtr_bufferSize()	49
	hipsolverDn<type>orgtr()	50
	hipsolverDn<type>ormqr_bufferSize()	50
	hipsolverDn<type>ormqr()	50
	hipsolverDn<type>ormtr_bufferSize()	50
	hipsolverDn<type>ormtr()	51
3.3.2	Unitary matrices	51
	hipsolverDn<type>ungbr_bufferSize()	51

	hipsolverDn<type>ungbr()	52
	hipsolverDn<type>ungqr_bufferSize()	52
	hipsolverDn<type>ungqr()	52
	hipsolverDn<type>ungtr_bufferSize()	52
	hipsolverDn<type>ungtr()	52
	hipsolverDn<type>unmqr_bufferSize()	53
	hipsolverDn<type>unmqr()	53
	hipsolverDn<type>unmtr_bufferSize()	53
	hipsolverDn<type>unmtr()	53
3.4	LAPACK Functions	54
3.4.1	Triangular factorizations	54
	hipsolverDn<type>potrf_bufferSize()	54
	hipsolverDn<type>potrf()	55
	hipsolverDn<type>potrfBatched()	55
	hipsolverDn<type>getrf_bufferSize()	55
	hipsolverDn<type>getrf()	55
	hipsolverDn<type>sytrf_bufferSize()	56
	hipsolverDn<type>sytrf()	56
3.4.2	Orthogonal factorizations	56
	hipsolverDn<type>geqrf_bufferSize()	56
	hipsolverDn<type>geqrf()	57
3.4.3	Problem and matrix reductions	57
	hipsolverDn<type>gebrd_bufferSize()	57
	hipsolverDn<type>gebrd()	57
	hipsolverDn<type>sytrd_bufferSize()	58
	hipsolverDn<type>hetrd_bufferSize()	58
	hipsolverDn<type>sytrd()	58
	hipsolverDn<type>hetrd()	58
3.4.4	Linear-systems solvers	59
	hipsolverDn<type>potri_bufferSize()	59
	hipsolverDn<type>potri()	59
	hipsolverDn<type>potrs()	60
	hipsolverDn<type>potrsBatched()	60
	hipsolverDn<type>getrs()	60
	hipsolverDn<type><type>gesv_bufferSize()	61
	hipsolverDn<type><type>gesv()	61
3.4.5	Least-squares solvers	61
	hipsolverDn<type><type>gels_bufferSize()	62
	hipsolverDn<type><type>gels()	62
3.4.6	Symmetric eigensolvers	62
	hipsolverDn<type>syevd_bufferSize()	63
	hipsolverDn<type>heevd_bufferSize()	63
	hipsolverDn<type>syevd()	63
	hipsolverDn<type>heevd()	63
	hipsolverDn<type>sygvd_bufferSize()	63
	hipsolverDn<type>hegvd_bufferSize()	64
	hipsolverDn<type>sygvd()	64
	hipsolverDn<type>hegvd()	64
3.4.7	Singular value decomposition	64
	hipsolverDn<type>gesvd_bufferSize()	65
	hipsolverDn<type>gesvd()	65
3.5	LAPACK-like Functions	65
3.5.1	Symmetric eigensolvers	65
	hipsolverDn<type>syevdx_bufferSize()	66

hipsolverDn<type>heevdx_bufferSize()	66
hipsolverDn<type>syevdx()	67
hipsolverDn<type>heevdx()	67
hipsolverDn<type>syevj_bufferSize()	67
hipsolverDn<type>heevj_bufferSize()	67
hipsolverDn<type>syevjBatched_bufferSize()	68
hipsolverDn<type>heevjBatched_bufferSize()	68
hipsolverDn<type>syevj()	68
hipsolverDn<type>heevj()	68
hipsolverDn<type>syevjBatched()	69
hipsolverDn<type>heevjBatched()	69
hipsolverDn<type>sygvdx_bufferSize()	69
hipsolverDn<type>hegvdx_bufferSize()	69
hipsolverDn<type>sygvdx()	70
hipsolverDn<type>hegvdx()	70
hipsolverDn<type>sygvj_bufferSize()	70
hipsolverDn<type>hegvj_bufferSize()	71
hipsolverDn<type>sygvj()	71
hipsolverDn<type>hegvj()	71
3.5.2 Singular value decomposition	71
hipsolverDn<type>gesvdj_bufferSize()	72
hipsolverDn<type>gesvdjBatched_bufferSize()	72
hipsolverDn<type>gesvdj()	73
hipsolverDn<type>gesvdjBatched()	73
hipsolverDn<type>gesvdaStridedBatched_bufferSize()	74
hipsolverDn<type>gesvdaStridedBatched()	75
Index	77

Legal Disclaimer

The information contained herein is for informational purposes only, and is subject to change without notice. In addition, any stated support is planned and is also subject to change. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

Contents

hipSOLVER's documentation consists of 3 main chapters. The User Guide is the starting point for new users of the library, and a basic reference for current users. For a list of currently implemented routines in the regular API, see the hipSOLVER API chapter. As explained in the User Guide, hipSOLVER also has an API that is fully compatible with cuSOLVER, described in the hipSOLVER Compatibility API document.

HIPSOLVER USER GUIDE

1.1 Introduction

Table of contents

- *Library overview*
- *Currently implemented functionality*
 - *LAPACK auxiliary functions*
 - *LAPACK main functions*
 - *LAPACK-like functions*
- *Compatibility-only functions*
 - *Partial eigensolver functions*
 - *Partial SVD functions*

1.1.1 Library overview

hipSOLVER is an open-source marshalling library for [LAPACK routines](#) on the GPU. It sits between a backend library and the user application, marshalling inputs to and outputs from the backend library so that the user application remains unchanged when using different backends. Currently, two backend libraries are supported by hipSOLVER: NVIDIA's [cuSOLVER library](#) and AMD's open-source [rocSOLVER library](#).

The *regular hipSOLVER API* is a thin wrapper layer around the different backends. As such, it is not expected to introduce significant overhead. However, its main purpose is portability, so when performance is critical, directly using the library backend corresponding to the given architecture is recommended.

Another purpose of hipSOLVER is to facilitate the translation of cuSOLVER applications to [AMD's open source ROCm platform ecosystem](#). To that end, hipSOLVER also includes a *compatibility API* with method signatures that match exactly those of cuSOLVER. For more details see the section [Using hipSOLVER](#).

1.1.2 Currently implemented functionality

The hipSOLVER library remains in active development. New features are being continuously added, with new functionality documented at each [release of the ROCm platform](#).

The following tables summarize the wrapper functions that are implemented in the regular API for the different supported precisions in hipSOLVER's latest release. Most of these functions have a corresponding version in the compatibility API, where applicable.

LAPACK auxiliary functions

Table 1: Orthonormal matrices

Function	single	double	single complex	double complex
<i>hipsolverXorgbr_bufferSize</i>	x	x		
<i>hipsolverXorgbr</i>	x	x		
<i>hipsolverXorgqr_bufferSize</i>	x	x		
<i>hipsolverXorgqr</i>	x	x		
<i>hipsolverXorgtr_bufferSize</i>	x	x		
<i>hipsolverXorgtr</i>	x	x		
<i>hipsolverXormqr_bufferSize</i>	x	x		
<i>hipsolverXormqr</i>	x	x		
<i>hipsolverXormtr_bufferSize</i>	x	x		
<i>hipsolverXormtr</i>	x	x		

Table 2: Unitary matrices

Function	single	double	single complex	double complex
<i>hipsolverXungbr_bufferSize</i>			x	x
<i>hipsolverXungbr</i>			x	x
<i>hipsolverXungqr_bufferSize</i>			x	x
<i>hipsolverXungqr</i>			x	x
<i>hipsolverXungtr_bufferSize</i>			x	x
<i>hipsolverXungtr</i>			x	x
<i>hipsolverXunmqr_bufferSize</i>			x	x
<i>hipsolverXunmqr</i>			x	x
<i>hipsolverXunmtr_bufferSize</i>			x	x
<i>hipsolverXunmtr</i>			x	x

LAPACK main functions

Table 3: Triangular factorizations

Function	single	double	single complex	double complex
<i>hipsolverXpotrf_bufferSize</i>	x	x	x	x
<i>hipsolverXpotrf</i>	x	x	x	x
<i>hipsolverXpotrfBatched_bufferSize</i>	x	x	x	x
<i>hipsolverXpotrfBatched</i>	x	x	x	x
<i>hipsolverXgetrf_bufferSize</i>	x	x	x	x
<i>hipsolverXgetrf</i>	x	x	x	x
<i>hipsolverXsytrf_bufferSize</i>	x	x	x	x
<i>hipsolverXsytrf</i>	x	x	x	x

Table 4: Orthogonal factorizations

Function	single	double	single complex	double complex
<i>hipsolverXgeqrf_bufferSize</i>	x	x	x	x
<i>hipsolverXgeqrf</i>	x	x	x	x

Table 5: Problem and matrix reductions

Function	single	double	single complex	double complex
<i>hipsolverXsytrd_bufferSize</i>	x	x		
<i>hipsolverXsytrd</i>	x	x		
<i>hipsolverXhetrd_bufferSize</i>			x	x
<i>hipsolverXhetrd</i>			x	x
<i>hipsolverXgebrd_bufferSize</i>	x	x	x	x
<i>hipsolverXgebrd</i>	x	x	x	x

Table 6: Linear-systems solvers

Function	single	double	single complex	double complex
<i>hipsolverXpotri_bufferSize</i>	x	x	x	x
<i>hipsolverXpotri</i>	x	x	x	x
<i>hipsolverXpotrs_bufferSize</i>	x	x	x	x
<i>hipsolverXpotrs</i>	x	x	x	x
<i>hipsolverXpotrsBatched_bufferSize</i>	x	x	x	x
<i>hipsolverXpotrsBatched</i>	x	x	x	x
<i>hipsolverXgetrs_bufferSize</i>	x	x	x	x
<i>hipsolverXgetrs</i>	x	x	x	x
<i>hipsolverXXgesv_bufferSize</i>	x	x	x	x
<i>hipsolverXXgesv</i>	x	x	x	x

Table 7: Least-square solvers

Function	single	double	single complex	double complex
<i>hipsolverXXgels_bufferSize</i>	x	x	x	x
<i>hipsolverXXgels</i>	x	x	x	x

Table 8: Symmetric eigensolvers

Function	single	double	single complex	double complex
<i>hipsolverXsyevd_bufferSize</i>	x	x		
<i>hipsolverXsyevd</i>	x	x		
<i>hipsolverXsygvd_bufferSize</i>	x	x		
<i>hipsolverXsygvd</i>	x	x		
<i>hipsolverXheevd_bufferSize</i>			x	x
<i>hipsolverXheevd</i>			x	x
<i>hipsolverXhegvd_bufferSize</i>			x	x
<i>hipsolverXhegvd</i>			x	x

Table 9: Singular value decomposition

Function	single	double	single complex	double complex
<i>hipsolverXgesvd_bufferSize</i>	x	x	x	x
<i>hipsolverXgesvd</i>	x	x	x	x

LAPACK-like functions

Table 10: Symmetric eigensolvers

Function	single	double	single complex	double complex
<i>hipsolverXsyevj_bufferSize</i>	x	x		
<i>hipsolverXsyevj</i>	x	x		
<i>hipsolverXsyevjBatched_bufferSize</i>	x	x		
<i>hipsolverXsyevjBatched</i>	x	x		
<i>hipsolverXsygvj_bufferSize</i>	x	x		
<i>hipsolverXsygvj</i>	x	x		
<i>hipsolverXheevj_bufferSize</i>			x	x
<i>hipsolverXheevj</i>			x	x
<i>hipsolverXheevjBatched_bufferSize</i>			x	x
<i>hipsolverXheevjBatched</i>			x	x
<i>hipsolverXhegvj_bufferSize</i>			x	x
<i>hipsolverXhegvj</i>			x	x

Table 11: Singular value decomposition

Function	single	double	single complex	double complex
<i>hipsolverDnXgesvdj_bufferSize</i>	x	x	x	x
<i>hipsolverDnXgesvdj</i>	x	x	x	x
<i>hipsolverDnXgesvdjBatched_bufferSize</i>	x	x	x	x
<i>hipsolverDnXgesvdjBatched</i>	x	x	x	x

1.1.3 Compatibility-only functions

The following tables summarize the wrapper functions that have not been implemented in rocSOLVER, but can be computed with an equivalent function that uses a different algorithmic approach. These functions are provided as part of the compatibility API only, pending their implementation in rocSOLVER.

Partial eigensolver functions

Partial eigensolvers have been implemented in rocSOLVER, but at present they do not use a divide-and-conquer approach.

Table 12: Symmetric eigensolvers

Function	single	double	single complex	double complex
<i>hipsolverDnXsyevdx_bufferSize</i>	x	x		
<i>hipsolverDnXsyevdx</i>	x	x		
<i>hipsolverDnXsygvdx_bufferSize</i>	x	x		
<i>hipsolverDnXsygvdx</i>	x	x		
<i>hipsolverDnXheevdx_bufferSize</i>			x	x
<i>hipsolverDnXheevdx</i>			x	x
<i>hipsolverDnXhegvdx_bufferSize</i>			x	x
<i>hipsolverDnXhegvdx</i>			x	x

Partial SVD functions

Partial SVD has been implemented in rocSOLVER, but at present it does not use an approximate algorithm, nor does it compute the residual norm.

Table 13: Singular value decomposition

Function	single	double	single complex	double complex
<i>hipsolverDnXgesvdaStridedBatched_bufferSize</i>	x	x	x	x
<i>hipsolverDnXgesvdaStridedBatched</i>	x	x	x	x

1.2 Installation

Table of contents

- *Install pre-built packages*
- *Build & install library using script (Ubuntu only)*
- *Build & install library manually*
- *Build library + tests + benchmarks + samples manually*
 - *Dependencies (only necessary for hipSOLVER clients)*
 - *Library and clients*

1.2.1 Install pre-built packages

Download pre-built packages from ROCm’s package servers. Release notes are available on the releases tab of the library’s github page.

- *sudo apt update && sudo apt install hipsolver*

1.2.2 Build & install library using script (Ubuntu only)

The root of the [hipSOLVER repository](#) has a helper bash script `install.sh` to build and install hipSOLVER on Ubuntu with a single command. It does not take a lot of options and hard-codes configuration that can be specified through invoking `cmake` directly, but it’s a great way to get started quickly and can serve as an example of how to build/install. A few commands in the script need `sudo` access, so it may prompt you for a password.

- `./install.sh -id` — build library, build dependencies, and install (`-d` flag only needs to be passed once on a system).
- `./install.sh -ic` — build library, build clients (tests, benchmarks, and samples), and install.

To see more options, use the `help` option of the install script.

- `./install.sh -h`

1.2.3 Build & install library manually

For a standard library installation, follow these steps:

```
mkdir -p <HIPSOLVER_BUILD_DIR_PATH>/release
cd <HIPSOLVER_BUILD_DIR_PATH>/release
CXX=/opt/rocm/bin/hipcc cmake <HIPSOLVER_SOURCE_DIR_PATH>
make -j$(nproc)
sudo make install
```

sudo is required if installing into a system directory such as /opt/rocm, which is the default option.

- Use `-DCMAKE_INSTALL_PREFIX=<other_path>` to specify a different install directory.
- Use `-DCMAKE_BUILD_TYPE=<other_configuration>` to specify a build configuration, such as 'Debug'. The default build configuration is 'Release'.

1.2.4 Build library + tests + benchmarks + samples manually

The repository contains source code for client programs that serve as tests, benchmarks, and samples. Client source code can be found in the clients subdirectory.

Dependencies (only necessary for hipSOLVER clients)

The hipSOLVER samples have no external dependencies, but our unit test and benchmarking applications do. These clients introduce the following dependencies:

1. `lapack` (lapack itself brings a dependency on a fortran compiler)
2. `googletest`

Unfortunately, many distros do not provide a googletest package with pre-compiled libraries, and the lapack packages do not have the necessary cmake config files for cmake to configure linking the cblas library. hipSOLVER provide a cmake script that builds the above dependencies from source. This is an optional step; users can provide their own builds of these dependencies and help cmake find them by setting the `CMAKE_PREFIX_PATH` definition. The following is a sequence of steps to build dependencies and install them to the cmake default, `/usr/local`:

```
mkdir -p <HIPSOLVER_BUILD_DIR_PATH>/release/deps
cd <HIPSOLVER_BUILD_DIR_PATH>/release/deps
cmake -DBUILD_BOOST=OFF <HIPSOLVER_SOURCE_PATH>/deps #assuming boost is installed,
↳through package manager as above
make -j$(nproc) install
```

Library and clients

Once dependencies are available on the system, it is possible to configure the clients to build. This requires a few extra cmake flags to the library's cmake configure script. If the dependencies are not installed into system defaults (like `/usr/local`), you should pass the `CMAKE_PREFIX_PATH` to cmake to help find them.

- `-DCMAKE_PREFIX_PATH="<semicolon separated paths>"`

```
CXX=/opt/rocm/bin/hipcc cmake -DBUILD_CLIENTS_TESTS=ON -DBUILD_CLIENTS_BENCHMARKS=ON
↳[HIPSOLVER_SOURCE]
make -j$(nproc)
sudo make install # sudo required if installing into system directory such as /opt/
↳rocm
```


1.3 Using hipSOLVER

Once installed, hipSOLVER can be used just like any other library with a C API. The header file will need to be included in the user code, and the shared library will become link-time and run-time dependencies for the user application. The user code can be ported, with no changes, to any system with hipSOLVER installed regardless of the backend library.

For more details on how to use the API methods, see the code samples on [hipSOLVER's github page](#), or the documentation of the corresponding backend libraries.

Table of contents

- *Porting cuSOLVER applications to hipSOLVER*
- *Some considerations when using the compatibility hipSOLVER API*
 - *Different minimum array lengths*
 - *Arguments not referenced by rocSOLVER*
 - *Possible performance implications of the compatibility API*
- *Some considerations when using the regular hipSOLVER API*
 - *Different signatures and additional API methods*
 - *Arguments not referenced by rocSOLVER*
 - *Using rocSOLVER's memory model*
 - *Using rocSOLVER's in-place functions*

1.3.1 Porting cuSOLVER applications to hipSOLVER

hipSOLVER is designed to make it easy for users of cuSOLVER to port their existing applications to hipSOLVER, and provides two separate but interchangeable APIs in order to facilitate a two-stage transition process. Users are encouraged to start with hipSOLVER's *compatibility API*, which uses the *hipsolverDn* prefix and has method signatures that are fully consistent with *cusolverDn* functions. However, the compatibility API may introduce some performance drawbacks, especially when using the rocSOLVER backend. So, as a second stage, it is recommended to begin the switch to hipSOLVER's *regular API*, which uses the *hipsolver* prefix and introduces minor adjustments to the API in order to get the best performance out of the rocSOLVER backend. In most cases, switching to the regular API is as simple as removing *Dn* from the *hipsolverDn* prefix. (No matter which API is used, a hipSOLVER application can be executed, without modifications to the code, in systems with cuSOLVER or rocSOLVER installed. However, using the regular API ensures the best performance out of both backends).

1.3.2 Some considerations when using the compatibility hipSOLVER API

hipSOLVER's compatibility API is intended as a 1:1 translation of the cuSOLVER API, but not all functionality is equally supported in rocSOLVER. Keep in mind the following considerations when using the compatibility API.

Different minimum array lengths

- Currently, the following methods require larger arrays than the minimum required by cuSOLVER.
 - *hipsolverDnXgesvdaStridedBatched* requires U to be of length $ldu * \min(m,n)$ at minimum, and S to be of length $\min(m,n)$ at minimum.

Arguments not referenced by rocSOLVER

- Unlike cuSOLVER, rocSOLVER functions do not provide information on invalid arguments in the *info* parameter, though they will provide info on singularities and algorithm convergence. Hence, when using the rocSOLVER backend, *info* will always return a value ≥ 0 . In those cases where a rocSOLVER function does not accept *info* as an argument, hipSOLVER will set it to zero.
- The *niters* argument of *hipsolverDnXXgels* and *hipsolverDnXXgesv* is not referenced by the rocSOLVER backend; there is no iterative refinement currently implemented in rocSOLVER.
- The *hRnmF* argument of *hipsolverDnXgesvdaStridedBatched* is not referenced by the rocSOLVER backend.

Possible performance implications of the compatibility API

- To calculate the workspace required by function *gesvd* in rocSOLVER, the values of *jobu* and *jobv* are needed, however, the function *hipsolverDnXgesvd_bufferSize* does not accept these arguments. So, when using the rocSOLVER backend, *hipsolverDnXgesvd_bufferSize* has to calculate internally the workspace for all possible values of *jobu* and *jobv*, and return the maximum.

(*hipsolverDnXgesvd_bufferSize* is slower than *hipsolverXgesvd_bufferSize*, and its returned workspace size could be slightly larger than what is actually needed).

- To properly use a user-provided workspace, rocSOLVER requires both the allocated pointer and its size. However, the function *hipsolverDnXgetrf* does not accept *lwork* as an argument. In consequence, when using the rocSOLVER backend, *hipsolverDnXgetrf* has to call internally *hipsolverDnXgetrf_bufferSize* to know the size of the workspace.

(*hipsolverDnXgetrf_bufferSize* will be called twice in practice, once by the user before allocating the workspace, and once by hipSOLVER internally when executing the *hipsolverDnXgetrf* function. *hipsolverDnXgetrf* could be slightly slower than *hipsolverXgetrf* because of the extra call to the *bufferSize* helper).

- The functions *hipsolverDnXgetrs*, *hipsolverDnXpotrs*, *hipsolverDnXpotrsBatched*, and *hipsolverDnXpotrfBatched* do not accept *work* and *lwork* as arguments. However, this functionality does require a non-zero workspace in rocSOLVER. As a result, when using the rocSOLVER backend, these functions will switch to the automatic workspace management model (see [here](#)).

(Users must keep in mind that even if the compatibility API does not have *bufferSize* helpers for the mentioned functions, these functions do require workspace when using rocSOLVER, and it will be automatically managed. This may imply device memory reallocations with corresponding overheads).

- The function *hipsolverDnXgesvdaStridedBatched* must apply a transpose operation to V in order to match the output of cuSOLVER, requiring an additional function call and extra workspace.

1.3.3 Some considerations when using the regular hipSOLVER API

hipSOLVER's regular API is similar to cuSOLVER; however, due to differences in the implementation and design between cuSOLVER and rocSOLVER, some minor adjustments were introduced to ensure the best performance out of both backends.

Different signatures and additional API methods

- The methods to obtain the size of the workspace needed by functions *gels* and *gesv* in cuSOLVER require *dwork* as an argument; however, it is never used and can be null. On the rocSOLVER side, *dwork* is not needed to calculate the workspace size. In consequence:
 - *hipsolverXXgels_bufferSize* does not require *dwork* as an argument, and
 - *hipsolverXXgesv_bufferSize* does not require *dwork* as an argument.
 (These wrappers pass *dwork = nullptr* when calling cuSOLVER).
- To calculate the workspace required by function *gesvd* in rocSOLVER, the values of *jobu* and *jobv* are needed. As a result,
 - *hipsolverXgesvd_bufferSize* requires *jobu* and *jobv* as arguments.
 (These arguments are ignored when the wrapper calls cuSOLVER, as they are not needed).
- To properly use a user-provided workspace, rocSOLVER requires both the allocated pointer and its size. Consequently:
 - *hipsolverXgetrf* requires *lwork* as an argument.
 (*lwork* is ignored when the wrapper calls cuSOLVER, as it is not needed).
- All rocSOLVER functions called by hipSOLVER require a workspace. To allow the user to specify one,
 - *hipsolverXgetrs* requires *work* and *lwork* as arguments,
 - *hipsolverXpotrfBatched* requires *work* and *lwork* as arguments,
 - *hipsolverXpotrs* requires *work* and *lwork* as arguments, and
 - *hipsolverXpotrsBatched* requires *work* and *lwork* as arguments.

(These arguments are ignored when these wrappers call cuSOLVER, as they are not needed).

In order to support these changes, the regular API adds the following functions as well:

- *hipsolverXgetrs_bufferSize*
- *hipsolverXpotrfBatched_bufferSize*
- *hipsolverXpotrs_bufferSize*
- *hipsolverXpotrsBatched_bufferSize*

(These methods return *lwork = 0* when using the cuSOLVER backend, as the corresponding functions in cuSOLVER do not need workspace).

Arguments not referenced by rocSOLVER

- Unlike cuSOLVER, rocSOLVER functions do not provide information on invalid arguments in the *info* parameter, though they will provide info on singularities and algorithm convergence. Hence, when using the rocSOLVER backend, *info* will always return a value ≥ 0 . In those cases where a rocSOLVER function does not accept *info* as an argument, hipSOLVER will set it to zero.
- The *niters* argument of *hipsolverXXgels* and *hipsolverXXgesv* is not referenced by the rocSOLVER backend; there is no iterative refinement currently implemented in rocSOLVER.

Using rocSOLVER's memory model

Most hipSOLVER functions take a workspace pointer and size as arguments, allowing the user to manage the device memory used internally by the backends. rocSOLVER, however, can maintain the device workspace automatically by default (see rocSOLVER's memory model for more details). In order to take advantage of this feature, users may pass a null pointer for the *work* argument or a zero size for the *lwork* argument of any function when using the rocSOLVER backend, and the workspace will be automatically managed behind-the-scenes. It is recommended, however, to use a consistent strategy for workspace management, as performance issues may arise if the internal workspace is made to flip-flop between user-provided and automatically allocated workspaces.

Warning: This feature should not be used with the cuSOLVER backend; hipSOLVER does not guarantee a defined behavior when passing a null workspace to cuSOLVER functions that require one.

Using rocSOLVER's in-place functions

The solvers *gesv* and *gels* in cuSOLVER are out-of-place in the sense that the solution vectors X do not overwrite the input matrix B . In rocSOLVER this is not the case; when *hipsolverXXgels* or *hipsolverXXgesv* call rocSOLVER, some data movements must be done internally to restore B and copy the results back to X . These copies could introduce noticeable overhead depending on the size of the matrices. To avoid this potential problem, users can pass $X = B$ to *hipsolverXXgels* or *hipsolverXXgesv* when using the rocSOLVER backend; in this case, no data movements will be required, and the solution vectors can be retrieved using either B or X .

Warning: This feature should not be used with the cuSOLVER backend; hipSOLVER does not guarantee a defined behavior when passing $X = B$ to the mentioned functions in cuSOLVER.

HIPSOLVER API

Currently, this API document only provides the method signatures for the wrapper functions that are currently implemented in hipSOLVER. For a complete description of the functions' behavior and arguments, see the corresponding backends' documentation at [cuSOLVER API](#) and/or [rocSOLVER API](#).

The hipSOLVER API is designed to be similar to the cusolverDn and rocSOLVER interfaces, but it requires some minor adjustments to ensure the best performance out of both backends. Generally, this involves the addition of workspace parameters and some additional API methods. Please refer to the user guide for a complete listing of *these API adjustments*.

Users interested in using hipSOLVER without these adjustments, so that the interface matches cuSOLVER, should instead consult the *compatibility API documentation*. See also *this section* for more details.

2.1 Types

hipSOLVER defines types and enumerations that are internally converted to cuBLAS/cuSOLVER or rocBLAS/rocSOLVER types at runtime. Here we list the types used in the regular API.

2.1.1 hipSOLVER regular API types

List of types in the regular API

- *hipsolverHandle_t*
- *hipsolverSyevjInfo_t*
- *hipsolverStatus_t*
- *hipsolverOperation_t*
- *hipsolverFillMode_t*
- *hipsolverSideMode_t*
- *hipsolverEigMode_t*
- *hipsolverEigType_t*

hipsolverHandle_t

```
typedef void *hipsolverHandle_t
```

hipsolverSyevjInfo_t

```
typedef void *hipsolverSyevjInfo_t
```

hipsolverStatus_t

```
enum hipsolverStatus_t
```

Values:

```
enumerator HIPSOLVER_STATUS_SUCCESS  
enumerator HIPSOLVER_STATUS_NOT_INITIALIZED  
enumerator HIPSOLVER_STATUS_ALLOC_FAILED  
enumerator HIPSOLVER_STATUS_INVALID_VALUE  
enumerator HIPSOLVER_STATUS_MAPPING_ERROR  
enumerator HIPSOLVER_STATUS_EXECUTION_FAILED  
enumerator HIPSOLVER_STATUS_INTERNAL_ERROR  
enumerator HIPSOLVER_STATUS_NOT_SUPPORTED  
enumerator HIPSOLVER_STATUS_ARCH_MISMATCH  
enumerator HIPSOLVER_STATUS_HANDLE_IS_NULLPTR  
enumerator HIPSOLVER_STATUS_INVALID_ENUM  
enumerator HIPSOLVER_STATUS_UNKNOWN
```

hipsolverOperation_t

```
enum hipsolverOperation_t
```

Values:

```
enumerator HIPSOLVER_OP_N  
enumerator HIPSOLVER_OP_T  
enumerator HIPSOLVER_OP_C
```

hipsolverFillMode_t

```
enum hipsolverFillMode_t
```

Values:

```
enumerator HIPSOLVER_FILL_MODE_UPPER  
enumerator HIPSOLVER_FILL_MODE_LOWER
```

hipsolverSideMode_t

enum hipsolverSideMode_t

Values:

enumerator HIPSOLVER_SIDE_LEFT

enumerator HIPSOLVER_SIDE_RIGHT

hipsolverEigMode_t

enum hipsolverEigMode_t

Values:

enumerator HIPSOLVER_EIG_MODE_NOVECTOR

enumerator HIPSOLVER_EIG_MODE_VECTOR

hipsolverEigType_t

enum hipsolverEigType_t

Values:

enumerator HIPSOLVER_EIG_TYPE_1

enumerator HIPSOLVER_EIG_TYPE_2

enumerator HIPSOLVER_EIG_TYPE_3

2.2 Helper Functions

These are helper functions that control aspects of the hipSOLVER library. These are divided into two categories:

- *Handle set-up and tear-down* functions. Used to initialize and cleanup the library handle.
- *Stream manipulation* functions. Provide functionality to manipulate streams.
- *Gesvdj parameter manipulation* functions. Provide functionality to manipulate gesvdj parameters.
- *Syevj parameter manipulation* functions. Provide functionality to manipulate syevj parameters.

2.2.1 Handle set-up and tear-down

List of handle initialization functions

- *hipsolverCreate()*
- *hipsolverDestroy()*

hipsolverCreate()

hipsolverStatus_t **hipsolverCreate** (*hipsolverHandle_t* *handle)

hipsolverDestroy()

hipsolverStatus_t **hipsolverDestroy** (*hipsolverHandle_t* handle)

2.2.2 Stream manipulation

List of stream manipulation functions

- *hipsolverSetStream*()
- *hipsolverGetStream*()

hipsolverSetStream()

hipsolverStatus_t **hipsolverSetStream** (*hipsolverHandle_t* handle, *hipStream_t* streamId)

hipsolverGetStream()

hipsolverStatus_t **hipsolverGetStream** (*hipsolverHandle_t* handle, *hipStream_t* *streamId)

2.2.3 Gesvdj parameter manipulation

List of gesvdj parameter functions

- *hipsolverCreateGesvdjInfo*()
- *hipsolverDestroyGesvdjInfo*()
- *hipsolverXgesvdjSetMaxSweeps*()
- *hipsolverXgesvdjSetSortEig*()
- *hipsolverXgesvdjSetTolerance*()
- *hipsolverXgesvdjGetResidual*()
- *hipsolverXgesvdjGetSweeps*()

hipsolverCreateGesvdjInfo()

hipsolverStatus_t **hipsolverCreateGesvdjInfo** (*hipsolverGesvdjInfo_t* **info*)

hipsolverDestroyGesvdjInfo()

hipsolverStatus_t **hipsolverDestroyGesvdjInfo** (*hipsolverGesvdjInfo_t* *info*)

hipsolverXgesvdjSetMaxSweeps()

hipsolverStatus_t **hipsolverXgesvdjSetMaxSweeps** (*hipsolverGesvdjInfo_t* *info*, int *max_sweeps*)

hipsolverXgesvdjSetSortEig()

hipsolverStatus_t **hipsolverXgesvdjSetSortEig** (*hipsolverGesvdjInfo_t* *info*, int *sort_eig*)

hipsolverXgesvdjSetTolerance()

hipsolverStatus_t **hipsolverXgesvdjSetTolerance** (*hipsolverGesvdjInfo_t* *info*, double *tolerance*)

hipsolverXgesvdjGetResidual()

hipsolverStatus_t **hipsolverXgesvdjGetResidual** (*hipsolverHandle_t* *handle*, *hipsolverGesvdjInfo_t* *info*, double **residual*)

hipsolverXgesvdjGetSweeps()

hipsolverStatus_t **hipsolverXgesvdjGetSweeps** (*hipsolverHandle_t* *handle*, *hipsolverGesvdjInfo_t* *info*, int **executed_sweeps*)

2.2.4 Syevj parameter manipulation

List of syevj parameter functions

- *hipsolverCreateSyevjInfo()*
- *hipsolverDestroySyevjInfo()*
- *hipsolverXsyevjSetMaxSweeps()*
- *hipsolverXsyevjSetSortEig()*
- *hipsolverXsyevjSetTolerance()*
- *hipsolverXsyevjGetResidual()*
- *hipsolverXsyevjGetSweeps()*

hipsolverCreateSyevjInfo()

hipsolverStatus_t **hipsolverCreateSyevjInfo** (*hipsolverSyevjInfo_t* *info)

hipsolverDestroySyevjInfo()

hipsolverStatus_t **hipsolverDestroySyevjInfo** (*hipsolverSyevjInfo_t* info)

hipsolverXsyevjSetMaxSweeps()

hipsolverStatus_t **hipsolverXsyevjSetMaxSweeps** (*hipsolverSyevjInfo_t* info, int max_sweeps)

hipsolverXsyevjSetSortEig()

hipsolverStatus_t **hipsolverXsyevjSetSortEig** (*hipsolverSyevjInfo_t* info, int sort_eig)

hipsolverXsyevjSetTolerance()

hipsolverStatus_t **hipsolverXsyevjSetTolerance** (*hipsolverSyevjInfo_t* info, double tolerance)

hipsolverXsyevjGetResidual()

hipsolverStatus_t **hipsolverXsyevjGetResidual** (*hipsolverHandle_t* handle, *hipsolverSyevjInfo_t* info, double *residual)

hipsolverXsyevjGetSweeps()

hipsolverStatus_t **hipsolverXsyevjGetSweeps** (*hipsolverHandle_t* handle, *hipsolverSyevjInfo_t* info, int *executed_sweeps)

2.3 LAPACK Auxiliary Functions

These are functions that support more *advanced LAPACK routines*. The auxiliary functions are divided into the following categories:

- *Orthonormal matrices*. Generation and application of orthonormal matrices.
- *Unitary matrices*. Generation and application of unitary matrices.

2.3.1 Orthonormal matrices

List of functions for orthonormal matrices

- *hipsolver<type>orgbr_bufferSize()*
- *hipsolver<type>orgbr()*
- *hipsolver<type>orgqr_bufferSize()*

- `hipsolver<type>orgqr()`
- `hipsolver<type>orgtr_bufferSize()`
- `hipsolver<type>orgtr()`
- `hipsolver<type>ormqr_bufferSize()`
- `hipsolver<type>ormqr()`
- `hipsolver<type>ormtr_bufferSize()`
- `hipsolver<type>ormtr()`

`hipsolver<type>orgbr_bufferSize()`

`hipsolverStatus_t hipsolverDorgbr_bufferSize` (`hipsolverHandle_t handle`, `hipsolverSideMode_t side`, `int m`, `int n`, `int k`, `double *A`, `int lda`, `double *tau`, `int *lwork`)

`hipsolverStatus_t hipsolverSorgbr_bufferSize` (`hipsolverHandle_t handle`, `hipsolverSideMode_t side`, `int m`, `int n`, `int k`, `float *A`, `int lda`, `float *tau`, `int *lwork`)

`hipsolver<type>orgbr()`

`hipsolverStatus_t hipsolverDorgbr` (`hipsolverHandle_t handle`, `hipsolverSideMode_t side`, `int m`, `int n`, `int k`, `double *A`, `int lda`, `double *tau`, `double *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverSorgbr` (`hipsolverHandle_t handle`, `hipsolverSideMode_t side`, `int m`, `int n`, `int k`, `float *A`, `int lda`, `float *tau`, `float *work`, `int lwork`, `int *devInfo`)

`hipsolver<type>orgqr_bufferSize()`

`hipsolverStatus_t hipsolverDorgqr_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int k`, `double *A`, `int lda`, `double *tau`, `int *lwork`)

`hipsolverStatus_t hipsolverSorgqr_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int k`, `float *A`, `int lda`, `float *tau`, `int *lwork`)

`hipsolver<type>orgqr()`

`hipsolverStatus_t hipsolverDorgqr` (`hipsolverHandle_t handle`, `int m`, `int n`, `int k`, `double *A`, `int lda`, `double *tau`, `double *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverSorgqr` (`hipsolverHandle_t handle`, `int m`, `int n`, `int k`, `float *A`, `int lda`, `float *tau`, `float *work`, `int lwork`, `int *devInfo`)

hipsolver<type>orgtr_bufferSize()

hipsolverStatus_t **hipsolverDorgtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *tau, int *lwork)

hipsolverStatus_t **hipsolverSorgtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *tau, int *lwork)

hipsolver<type>orgtr()

hipsolverStatus_t **hipsolverDorgtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSorgtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *tau, float *work, int lwork, int *devInfo)

hipsolver<type>ormqr_bufferSize()

hipsolverStatus_t **hipsolverDormqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, double *A, int lda, double *tau, double *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverSormqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, float *A, int lda, float *tau, float *C, int ldc, int *lwork)

hipsolver<type>ormqr()

hipsolverStatus_t **hipsolverDormqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, double *A, int lda, double *tau, double *C, int ldc, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSormqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, float *A, int lda, float *tau, float *C, int ldc, float *work, int lwork, int *devInfo)

hipsolver<type>ormtr_bufferSize()

hipsolverStatus_t **hipsolverDormtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, double *A, int lda, double *tau, double *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverSormtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, float *A, int lda, float *tau, float *C, int ldc, int *lwork)

hipsolver<type>ormtr()

hipsolverStatus_t **hipsolverDormtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, double *A, int lda, double *tau, double *C, int ldc, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSormtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, float *A, int lda, float *tau, float *C, int ldc, float *work, int lwork, int *devInfo)

2.3.2 Unitary matrices**List of functions for unitary matrices**

- *hipsolver<type>ungbr_bufferSize()*
- *hipsolver<type>ungbr()*
- *hipsolver<type>ungqr_bufferSize()*
- *hipsolver<type>ungqr()*
- *hipsolver<type>ungtr_bufferSize()*
- *hipsolver<type>ungtr()*
- *hipsolver<type>unmqr_bufferSize()*
- *hipsolver<type>unmqr()*
- *hipsolver<type>unmtr_bufferSize()*
- *hipsolver<type>unmtr()*

hipsolver<type>ungbr_bufferSize()

hipsolverStatus_t **hipsolverZungbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverCungbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, int *lwork)

hipsolver<type>ungbr()

hipsolverStatus_t **hipsolverZungbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCungbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolver<type>ungqr_bufferSize()

hipsolverStatus_t **hipsolverZungqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverCungqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, int *lwork)

hipsolver<type>ungqr()

hipsolverStatus_t **hipsolverZungqr** (*hipsolverHandle_t* handle, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCungqr** (*hipsolverHandle_t* handle, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolver<type>ungtr_bufferSize()

hipsolverStatus_t **hipsolverZungtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverCungtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, int *lwork)

hipsolver<type>ungtr()

hipsolverStatus_t **hipsolverZungtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCungtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolver<type>unmqr_bufferSize()

hipsolverStatus_t **hipsolverZunmqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverCunmqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *C, int ldc, int *lwork)

hipsolver<type>unmqr()

hipsolverStatus_t **hipsolverZunmqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCunmqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *C, int ldc, hipFloatComplex *work, int lwork, int *devInfo)

hipsolver<type>unmtr_bufferSize()

hipsolverStatus_t **hipsolverZunmtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverCunmtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *C, int ldc, int *lwork)

hipsolver<type>unmtr()

hipsolverStatus_t **hipsolverZunmtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCunmtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *C, int ldc, hipFloatComplex *work, int lwork, int *devInfo)

2.4 LAPACK Functions

LAPACK routines solve complex Numerical Linear Algebra problems. These functions are organized in the following categories:

- *Triangular factorizations.* Based on Gaussian elimination.
- *Orthogonal factorizations.* Based on Householder reflections.
- *Problem and matrix reductions.* Transformation of matrices and problems into equivalent forms.
- *Linear-systems solvers.* Based on triangular factorizations.
- *Least-squares solvers.* Based on orthogonal factorizations.
- *Symmetric eigensolvers.* Eigenproblems for symmetric matrices.
- *Singular value decomposition.* Singular values and related problems for general matrices.

2.4.1 Triangular factorizations

List of triangular factorizations

- `hipsolver<type>potrf_bufferSize()`
- `hipsolver<type>potrfBatched_bufferSize()`
- `hipsolver<type>potrf()`
- `hipsolver<type>potrfBatched()`
- `hipsolver<type>getrf_bufferSize()`
- `hipsolver<type>getrf()`
- `hipsolver<type>sytrf_bufferSize()`
- `hipsolver<type>sytrf()`

`hipsolver<type>potrf_bufferSize()`

`hipsolverStatus_t hipsolverZpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipDoubleComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverCpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipFloatComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverSpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `int *lwork`)

hipsolver<type>potrfBatched_bufferSize()

hipsolverStatus_t **hipsolverZpotrfBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A[], int lda, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverCpotrfBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A[], int lda, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverDpotrfBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A[], int lda, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverSpotrfBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A[], int lda, int *lwork, int batch_count)

hipsolver<type>potrf()

hipsolverStatus_t **hipsolverZpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *work, int lwork, int *devInfo)

hipsolver<type>potrfBatched()

hipsolverStatus_t **hipsolverZpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A[], int lda, hipDoubleComplex *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverCpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A[], int lda, hipFloatComplex *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A[], int lda, double *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverSpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A[], int lda, float *work, int lwork, int *devInfo, int batch_count)

hipsolver<type>getrf_bufferSize()

hipsolverStatus_t **hipsolverZgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverCgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverSgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, int *lwork)

hipsolver<type>getrf()

hipsolverStatus_t **hipsolverZgetrf** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *work, int lwork, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverCgetrf** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *work, int lwork, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverDgetrf** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, double *work, int lwork, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverSgetrf** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, float *work, int lwork, int *devI piv, int *devInfo)

hipsolver<type>sytrf_bufferSize()

hipsolverStatus_t **hipsolverZsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverCsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverSsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, float *A, int lda, int *lwork)

hipsolver<type>sytrf()

hipsolverStatus_t **hipsolverZsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, int *ipiv, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, int *ipiv, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, int *ipiv, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, int *ipiv, float *work, int lwork, int *devInfo)

2.4.2 Orthogonal factorizations

List of orthogonal factorizations

- `hipsolver<type>geqrf_bufferSize()`
- `hipsolver<type>geqrf()`

`hipsolver<type>geqrf_bufferSize()`

`hipsolverStatus_t hipsolverZgeqrf_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `hipDoubleComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverCgeqrf_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `hipFloatComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDgeqrf_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `double *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverSgeqrf_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `float *A`, `int lda`, `int *lwork`)

`hipsolver<type>geqrf()`

`hipsolverStatus_t hipsolverZgeqrf` (`hipsolverHandle_t handle`, `int m`, `int n`, `hipDoubleComplex *A`, `int lda`, `hipDoubleComplex *tau`, `hipDoubleComplex *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverCgeqrf` (`hipsolverHandle_t handle`, `int m`, `int n`, `hipFloatComplex *A`, `int lda`, `hipFloatComplex *tau`, `hipFloatComplex *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverDgeqrf` (`hipsolverHandle_t handle`, `int m`, `int n`, `double *A`, `int lda`, `double *tau`, `double *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverSgeqrf` (`hipsolverHandle_t handle`, `int m`, `int n`, `float *A`, `int lda`, `float *tau`, `float *work`, `int lwork`, `int *devInfo`)

2.4.3 Problem and matrix reductions

List of reductions

- `hipsolver<type>gebrd_bufferSize()`
- `hipsolver<type>gebrd()`
- `hipsolver<type>sytrd_bufferSize()`
- `hipsolver<type>hetrd_bufferSize()`
- `hipsolver<type>sytrd()`
- `hipsolver<type>hetrd()`

hipsolver<type>gebrd_bufferSize()

hipsolverStatus_t **hipsolverZgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverCgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverSgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolver<type>gebrd()

hipsolverStatus_t **hipsolverZgebrd** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, double *D, double *E, hipDoubleComplex *tauq, hipDoubleComplex *taup, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCgebrd** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, float *D, float *E, hipFloatComplex *tauq, hipFloatComplex *taup, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDgebrd** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, double *D, double *E, double *tauq, double *taup, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSgebrd** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, float *D, float *E, float *tauq, float *taup, float *work, int lwork, int *devInfo)

hipsolver<type>sytrd_bufferSize()

hipsolverStatus_t **hipsolverDsytrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *D, double *E, double *tau, int *lwork)

hipsolverStatus_t **hipsolverSsytrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *D, float *E, float *tau, int *lwork)

hipsolver<type>hetrd_bufferSize()

hipsolverStatus_t **hipsolverZhetrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *D, double *E, hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverChetrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *D, float *E, hipFloatComplex *tau, int *lwork)

hipsolver<type>sytrd()

hipsolverStatus_t **hipsolverDsytrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *D, double *E, double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSsytrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *D, float *E, float *tau, float *work, int lwork, int *devInfo)

hipsolver<type>hetrd()

hipsolverStatus_t **hipsolverZhetrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *D, double *E, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverChetrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *D, float *E, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

2.4.4 Linear-systems solvers**List of linear solvers**

- *hipsolver<type>potri_bufferSize()*
- *hipsolver<type>potri()*
- *hipsolver<type>potrs_bufferSize()*
- *hipsolver<type>potrsBatched_bufferSize()*
- *hipsolver<type>potrs()*
- *hipsolver<type>potrsBatched()*
- *hipsolver<type>getrs_bufferSize()*
- *hipsolver<type>getrs()*
- *hipsolver<type><type>gesv_bufferSize()*
- *hipsolver<type><type>gesv()*

hipsolver<type>potri_bufferSize()

hipsolverStatus_t **hipsolverZpotri_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverCpotri_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDpotri_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverSpotri_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, int *lwork)

hipsolver<type>potri()

hipsolverStatus_t **hipsolverZpotri** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCpotri** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDpotri** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSpotri** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *work, int lwork, int *devInfo)

hipsolver<type>potrs_bufferSize()

hipsolverStatus_t **hipsolverZpotrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverCpotrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverDpotrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, double *A, int lda, double *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverSpotrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, float *A, int lda, float *B, int ldb, int *lwork)

hipsolver<type>potrsBatched_bufferSize()

hipsolverStatus_t **hipsolverZpotrsBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipDoubleComplex *A[], int lda, hipDoubleComplex *B[], int ldb, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverCpotrsBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipFloatComplex *A[], int lda, hipFloatComplex *B[], int ldb, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverDpotrsBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, double *A[], int lda, double *B[], int ldb, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverSpotrsBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, float *A[], int lda, float *B[], int ldb, int *lwork, int batch_count)

hipsolver<type>potrs()

hipsolverStatus_t **hipsolverZpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, double *A, int lda, double *B, int ldb, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, float *A, int lda, float *B, int ldb, float *work, int lwork, int *devInfo)

hipsolver<type>potrsBatched()

hipsolverStatus_t **hipsolverZpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipDoubleComplex *A[], int lda, hipDoubleComplex *B[], int ldb, hipDoubleComplex *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverCpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipFloatComplex *A[], int lda, hipFloatComplex *B[], int ldb, hipFloatComplex *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, double *A[], int lda, double *B[], int ldb, double *work, int lwork, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverSpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, float *A[], int lda, float *B[], int ldb, float *work, int lwork, int *devInfo, int batch_count)

hipsolver<type>getrs_bufferSize()

hipsolverStatus_t **hipsolverZgetrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, hipDoubleComplex *A, int lda, int *devI piv, hipDoubleComplex *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverCgetrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, hipFloatComplex *A, int lda, int *devI piv, hipFloatComplex *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverDgetrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, double *A, int lda, int *devI piv, double *B, int ldb, int *lwork)

hipsolverStatus_t **hipsolverSgetrs_bufferSize** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, float *A, int lda, int *devI piv, float *B, int ldb, int *lwork)

hipsolver<type>getrs()

hipsolverStatus_t **hipsolverZgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, hipDoubleComplex *A, int lda, int *devIpivot, hipDoubleComplex *B, int ldb, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, hipFloatComplex *A, int lda, int *devIpivot, hipFloatComplex *B, int ldb, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, double *A, int lda, int *devIpivot, double *B, int ldb, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, float *A, int lda, int *devIpivot, float *B, int ldb, float *work, int lwork, int *devInfo)

hipsolver<type><type>gesv_bufferSize()

hipsolverStatus_t **hipsolverZZgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, hipDoubleComplex *A, int lda, int *devIpivot, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, size_t *lwork)

hipsolverStatus_t **hipsolverCCgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, hipFloatComplex *A, int lda, int *devIpivot, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, size_t *lwork)

hipsolverStatus_t **hipsolverDDgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, double *A, int lda, int *devIpivot, double *B, int ldb, double *X, int ldx, size_t *lwork)

hipsolverStatus_t **hipsolverSSgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, float *A, int lda, int *devIpivot, float *B, int ldb, float *X, int ldx, size_t *lwork)

hipsolver<type><type>gesv()

hipsolverStatus_t **hipsolverZZgesv** (*hipsolverHandle_t* handle, int n, int nrhs, hipDoubleComplex *A, int lda, int *devIpivot, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverCCgesv** (*hipsolverHandle_t* handle, int n, int nrhs, hipFloatComplex *A, int lda, int *devIpivot, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverDDgesv** (*hipsolverHandle_t* handle, int n, int nrhs, double *A, int lda, int *devIpivot, double *B, int ldb, double *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverSSgesv** (*hipsolverHandle_t* handle, int n, int nrhs, float *A, int lda, int *devIpivot, float *B, int ldb, float *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

2.4.5 Least-squares solvers

List of least-squares solvers

- `hipsolver<type><type>gels_bufferSize()`
- `hipsolver<type><type>gels()`

`hipsolver<type><type>gels_bufferSize()`

`hipsolverStatus_t hipsolverZZgels_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `hipDoubleComplex *A`, `int lda`, `hipDoubleComplex *B`, `int ldb`, `hipDoubleComplex *X`, `int ldx`, `size_t *lwork`)

`hipsolverStatus_t hipsolverCCgels_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `hipFloatComplex *A`, `int lda`, `hipFloatComplex *B`, `int ldb`, `hipFloatComplex *X`, `int ldx`, `size_t *lwork`)

`hipsolverStatus_t hipsolverDDgels_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `double *A`, `int lda`, `double *B`, `int ldb`, `double *X`, `int ldx`, `size_t *lwork`)

`hipsolverStatus_t hipsolverSSgels_bufferSize` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `float *A`, `int lda`, `float *B`, `int ldb`, `float *X`, `int ldx`, `size_t *lwork`)

`hipsolver<type><type>gels()`

`hipsolverStatus_t hipsolverZZgels` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `hipDoubleComplex *A`, `int lda`, `hipDoubleComplex *B`, `int ldb`, `hipDoubleComplex *X`, `int ldx`, `void *work`, `size_t lwork`, `int *nitters`, `int *devInfo`)

`hipsolverStatus_t hipsolverCCgels` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `hipFloatComplex *A`, `int lda`, `hipFloatComplex *B`, `int ldb`, `hipFloatComplex *X`, `int ldx`, `void *work`, `size_t lwork`, `int *nitters`, `int *devInfo`)

`hipsolverStatus_t hipsolverDDgels` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `double *A`, `int lda`, `double *B`, `int ldb`, `double *X`, `int ldx`, `void *work`, `size_t lwork`, `int *nitters`, `int *devInfo`)

`hipsolverStatus_t hipsolverSSgels` (`hipsolverHandle_t handle`, `int m`, `int n`, `int nrhs`, `float *A`, `int lda`, `float *B`, `int ldb`, `float *X`, `int ldx`, `void *work`, `size_t lwork`, `int *nitters`, `int *devInfo`)

2.4.6 Symmetric eigensolvers

List of symmetric eigensolvers

- `hipsolver<type>syevd_bufferSize()`
- `hipsolver<type>heevd_bufferSize()`
- `hipsolver<type>syevd()`
- `hipsolver<type>heevd()`
- `hipsolver<type>sygvd_bufferSize()`
- `hipsolver<type>hegvd_bufferSize()`
- `hipsolver<type>sygvd()`
- `hipsolver<type>hegvd()`

`hipsolver<type>syevd_bufferSize()`

`hipsolverStatus_t hipsolverDsyevd_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `double *D`, `int *lwork`)

`hipsolverStatus_t hipsolverSsyevd_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `float *D`, `int *lwork`)

`hipsolver<type>heevd_bufferSize()`

`hipsolverStatus_t hipsolverZheevd_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `hipDoubleComplex *A`, `int lda`, `double *D`, `int *lwork`)

`hipsolverStatus_t hipsolverCheevd_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `hipFloatComplex *A`, `int lda`, `float *D`, `int *lwork`)

`hipsolver<type>syevd()`

`hipsolverStatus_t hipsolverDsyevd` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `double *D`, `double *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverSsyevd` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `float *D`, `float *work`, `int lwork`, `int *devInfo`)

hipsolver<type>heevd()

hipsolverStatus_t **hipsolverZheevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *D, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverCheevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *D, hipFloatComplex *work, int lwork, int *devInfo)

hipsolver<type>sygvd_bufferSize()

hipsolverStatus_t **hipsolverDsygvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, int *lwork)

hipsolverStatus_t **hipsolverSsygvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, int *lwork)

hipsolver<type>hegvd_bufferSize()

hipsolverStatus_t **hipsolverZhegvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, int *lwork)

hipsolverStatus_t **hipsolverChegvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, int *lwork)

hipsolver<type>sygvd()

hipsolverStatus_t **hipsolverDsygvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverSsygvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, float *work, int lwork, int *devInfo)

hipsolver<type>hegvd()

hipsolverStatus_t **hipsolverZhegvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverChegvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, hipFloatComplex *work, int lwork, int *devInfo)

2.4.7 Singular value decomposition**List of SVD related functions**

- *hipsolver<type>gesvd_bufferSize()*
- *hipsolver<type>gesvd()*

hipsolver<type>gesvd_bufferSize()

hipsolverStatus_t **hipsolverZgesvd_bufferSize** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverCgesvd_bufferSize** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDgesvd_bufferSize** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverSgesvd_bufferSize** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, int *lwork)

hipsolver<type>gesvd()

hipsolverStatus_t **hipsolverZgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, double *rwork, int *devInfo)

hipsolverStatus_t **hipsolverCgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, float *rwork, int *devInfo)

hipsolverStatus_t **hipsolverDgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, double *rwork, int *devInfo)

hipsolverStatus_t **hipsolverSgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, float *rwork, int *devInfo)

2.5 LAPACK-like Functions

Other Lapack-like routines provided by hipSOLVER. These are divided into the following subcategories:

- *Symmetric eigensolvers*. Eigenproblems for symmetric matrices.
- *Singular value decomposition*. Singular values and related problems for general matrices.

2.5.1 Symmetric eigensolvers

List of Lapack-like symmetric eigensolvers

- `hipsolver<type>syevj_bufferSize()`
- `hipsolver<type>heevj_bufferSize()`
- `hipsolver<type>syevjBatched_bufferSize()`
- `hipsolver<type>heevjBatched_bufferSize()`
- `hipsolver<type>syevj()`
- `hipsolver<type>heevj()`
- `hipsolver<type>syevjBatched()`
- `hipsolver<type>heevjBatched()`
- `hipsolver<type>sygvj_bufferSize()`
- `hipsolver<type>hegvj_bufferSize()`
- `hipsolver<type>sygvj()`
- `hipsolver<type>hegvj()`

`hipsolver<type>syevj_bufferSize()`

`hipsolverStatus_t hipsolverDsyevj_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `double *W`, `int *lwork`, `hipsolverSyevjInfo_t params`)

`hipsolverStatus_t hipsolverSsyevj_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `float *W`, `int *lwork`, `hipsolverSyevjInfo_t params`)

hipsolver<type>heevj_bufferSize()

hipsolverStatus_t **hipsolverZheevj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverCheevj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolver<type>syevjBatched_bufferSize()

hipsolverStatus_t **hipsolverDsyevjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverSsyevjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolver<type>heevjBatched_bufferSize()

hipsolverStatus_t **hipsolverZheevjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverCheevjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolver<type>syevj()

hipsolverStatus_t **hipsolverDsyevj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, double *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverSsyevj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, float *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolver<type>heevj()

hipsolverStatus_t **hipsolverZheevj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

hipsolverStatus_t **hipsolverCheevj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

hipsolver<type>syejBatched()

hipsolverStatus_t **hipsolverDsyejBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, double *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverSsyejBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, float *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params, int batch_count)

hipsolver<type>heevjBatched()

hipsolverStatus_t **hipsolverZheevjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverCheevjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params, int batch_count)

hipsolver<type>sygvj_bufferSize()

hipsolverStatus_t **hipsolverDsygvj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, int *lwork, *hipsolverSyejInfo_t* params)

hipsolverStatus_t **hipsolverSsygvj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, int *lwork, *hipsolverSyejInfo_t* params)

hipsolver<type>hegvj_bufferSize()

hipsolverStatus_t **hipsolverZhegvj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverChegvj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolver<type>sygvj()

hipsolverStatus_t **hipsolverDsygvj** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, double *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverSsygvj** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, float *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolver<type>hegvj()

hipsolverStatus_t **hipsolverZhegvj** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverChegvj** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

2.5.2 Singular value decomposition**List of Lapack-like SVD related functions**

- *hipsolver<type>gesvdj_bufferSize()*
- *hipsolver<type>gesvdjBatched_bufferSize()*
- *hipsolver<type>gesvdj()*
- *hipsolver<type>gesvdjBatched()*

hipsolver<type>gesvdj_bufferSize()

hipsolverStatus_t **hipsolverZgesvdj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, **const** hipDoubleComplex *A, int lda, **const** double *S, **const** hipDoubleComplex *U, int ldu, **const** hipDoubleComplex *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverCgesvdj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, **const** hipFloatComplex *A, int lda, **const** float *S, **const** hipFloatComplex *U, int ldu, **const** hipFloatComplex *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverDgesvdj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, **const** double *A, int lda, **const** double *S, **const** double *U, int ldu, **const** double *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverSgesvdj_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, **const** float *A, int lda, **const** float *S, **const** float *U, int ldu, **const** float *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params)

hipsolver<type>gesvdjBatched_bufferSize()

hipsolverStatus_t **hipsolverZgesvdjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, **const** hipDoubleComplex *A, int lda, **const** double *S, **const** hipDoubleComplex *U, int ldu, **const** hipDoubleComplex *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverCgesvdjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, **const** hipFloatComplex *A, int lda, **const** float *S, **const** hipFloatComplex *U, int ldu, **const** hipFloatComplex *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDgesvdjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, **const** double *A, int lda, **const** double *S, **const** double *U, int ldu, **const** double *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverSgesvdjBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, const float *A, int lda, const float *S, const float *U, int ldu, const float *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolver<type>gesvdj()

hipsolverStatus_t **hipsolverZgesvdj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverCgesvdj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverDgesvdj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverSgesvdj** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolver<type>gesvdjBatched()

hipsolverStatus_t **hipsolverZgesvdjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverCgesvdjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDgesvdjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverSgesvdjBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

HIPSOLVER COMPATIBILITY API

Currently, this API document only provides the method signatures for the wrapper functions that are currently implemented in hipSOLVER. For a complete description of the functions' behavior and arguments, see the corresponding backends' documentation at [cuSOLVER API](#) and/or [rocSOLVER API](#).

For ease of porting from existing cuSOLVER applications to hipSOLVER, functions in hipSOLVER's compatibility API are designed to have method signatures that are consistent with the cusolverDn interface. However, *performance issues* may arise when using the rocSOLVER backend due to differing workspace requirements. Therefore, users interested in achieving the best performance with the rocSOLVER backend should consult the *regular API documentation*, and transition from the compatibility API to the regular API at the earliest convenience. Please refer to the user guide for additional *considerations regarding the use of the compatibility API*.

3.1 Types

hipSOLVER defines types and enumerations that are internally converted to cuBLAS/cuSOLVER or rocBLAS/rocSOLVER types at runtime. Here we list the types used in the compatibility API.

3.1.1 hipSOLVER compatibility API types

List of types in the compatibility API

- *hipsolverDnHandle_t*
- *hipsolverGesvdjInfo_t*
- *hipsolverSyevjInfo_t*
- *hipsolverStatus_t*
- *hipsolverOperation_t*
- *hipsolverFillMode_t*
- *hipsolverSideMode_t*
- *hipsolverEigMode_t*
- *hipsolverEigType_t*
- *hipsolverEigRange_t*

hipsolverDnHandle_t

typedef *hipsolverHandle_t* **hipsolverDnHandle_t**
Provided for convenience when porting code from cuSOLVER.

hipsolverGesvdjInfo_t

typedef void ***hipsolverGesvdjInfo_t**

hipsolverSyevjInfo_t

See *hipsolverSyevjInfo_t*.

hipsolverStatus_t

See *hipsolverStatus_t*.

hipsolverOperation_t

See *hipsolverOperation_t*.

hipsolverFillMode_t

See *hipsolverFillMode_t*.

hipsolverSideMode_t

See *hipsolverSideMode_t*.

hipsolverEigMode_t

See *hipsolverEigMode_t*.

hipsolverEigType_t

See *hipsolverEigType_t*.

hipsolverEigRange_t

enum **hipsolverEigRange_t**
Values:
enumerator **HIPSOLVER_EIG_RANGE_ALL**
enumerator **HIPSOLVER_EIG_RANGE_V**
enumerator **HIPSOLVER_EIG_RANGE_I**

3.2 Helper Functions

These are helper functions that control aspects of the hipSOLVER library. These are divided into two categories:

- *Handle set-up and tear-down* functions. Used to initialize and cleanup the library handle.
- *Stream manipulation* functions. Provide functionality to manipulate streams.
- *Gesvdj parameter manipulation* functions. Provide functionality to manipulate gesvdj parameters.
- *Syevj parameter manipulation* functions. Provide functionality to manipulate syevj parameters.

3.2.1 Handle set-up and tear-down

List of handle initialization functions

- *hipsolverDnCreate()*
- *hipsolverDnDestroy()*

hipsolverDnCreate()

hipsolverStatus_t **hipsolverDnCreate** (*hipsolverHandle_t* *handle)

An alias for *hipsolverCreate*.

hipsolverDnDestroy()

hipsolverStatus_t **hipsolverDnDestroy** (*hipsolverHandle_t* handle)

An alias for *hipsolverDestroy*.

3.2.2 Stream manipulation

List of stream manipulation functions

- *hipsolverDnSetStream()*
- *hipsolverDnGetStream()*

hipsolverDnSetStream()

hipsolverStatus_t **hipsolverDnSetStream** (*hipsolverHandle_t* handle, *hipStream_t* streamId)

An alias for *hipsolverSetStream*.

hipsolverDnGetStream()

hipsolverStatus_t **hipsolverDnGetStream** (*hipsolverHandle_t* handle, *hipStream_t* *streamId)
An alias for *hipsolverGetStream*.

3.2.3 Gesvdj parameter manipulation

List of gesvdj parameter functions

- *hipsolverDnCreateGesvdjInfo*()
- *hipsolverDnDestroyGesvdjInfo*()
- *hipsolverDnXgesvdjSetMaxSweeps*()
- *hipsolverDnXgesvdjSetSortEig*()
- *hipsolverDnXgesvdjSetTolerance*()
- *hipsolverDnXgesvdjGetResidual*()
- *hipsolverDnXgesvdjGetSweeps*()

hipsolverDnCreateGesvdjInfo()

hipsolverStatus_t **hipsolverDnCreateGesvdjInfo** (*hipsolverGesvdjInfo_t* *info)

hipsolverDnDestroyGesvdjInfo()

hipsolverStatus_t **hipsolverDnDestroyGesvdjInfo** (*hipsolverGesvdjInfo_t* info)

hipsolverDnXgesvdjSetMaxSweeps()

hipsolverStatus_t **hipsolverDnXgesvdjSetMaxSweeps** (*hipsolverGesvdjInfo_t* info, int max_sweeps)

hipsolverDnXgesvdjSetSortEig()

hipsolverStatus_t **hipsolverDnXgesvdjSetSortEig** (*hipsolverGesvdjInfo_t* info, int sort_eig)

hipsolverDnXgesvdjSetTolerance()

hipsolverStatus_t **hipsolverDnXgesvdjSetTolerance** (*hipsolverGesvdjInfo_t* info, double tolerance)

hipsolverDnXgesvdjGetResidual()

hipsolverStatus_t **hipsolverDnXgesvdjGetResidual** (*hipsolverDnHandle_t* handle, *hipsolverGesvdjInfo_t* info, double *residual)

hipsolverDnXgesvdjGetSweeps()

hipsolverStatus_t **hipsolverDnXgesvdjGetSweeps** (*hipsolverDnHandle_t* handle, *hipsolverGesvdjInfo_t* info, int *executed_sweeps)

3.2.4 Syevj parameter manipulation**List of syevj parameter functions**

- *hipsolverDnCreateSyevjInfo()*
- *hipsolverDnDestroySyevjInfo()*
- *hipsolverDnXsyevjSetMaxSweeps()*
- *hipsolverDnXsyevjSetSortEig()*
- *hipsolverDnXsyevjSetTolerance()*
- *hipsolverDnXsyevjGetResidual()*
- *hipsolverDnXsyevjGetSweeps()*

hipsolverDnCreateSyevjInfo()

hipsolverStatus_t **hipsolverDnCreateSyevjInfo** (*hipsolverSyevjInfo_t* *info)

hipsolverDnDestroySyevjInfo()

hipsolverStatus_t **hipsolverDnDestroySyevjInfo** (*hipsolverSyevjInfo_t* info)

hipsolverDnXsyevjSetMaxSweeps()

hipsolverStatus_t **hipsolverDnXsyevjSetMaxSweeps** (*hipsolverSyevjInfo_t* info, int max_sweeps)

hipsolverDnXsyevjSetSortEig()

hipsolverStatus_t **hipsolverDnXsyevjSetSortEig** (*hipsolverSyevjInfo_t* info, int sort_eig)

hipsolverDnXsyevjSetTolerance()

hipsolverStatus_t **hipsolverDnXsyevjSetTolerance** (*hipsolverSyevjInfo_t* info, double tolerance)

hipsolverDnXsyevjGetResidual()

hipsolverStatus_t **hipsolverDnXsyevjGetResidual** (*hipsolverDnHandle_t* handle, *hipsolverSyevjInfo_t* info, double *residual)

hipsolverDnXsyevjGetSweeps()

hipsolverStatus_t **hipsolverDnXsyevjGetSweeps** (*hipsolverDnHandle_t* handle, *hipsolverSyevjInfo_t* info, int *executed_sweeps)

3.3 LAPACK Auxiliary Functions

These are functions that support more *advanced LAPACK routines*. The auxiliary functions are divided into the following categories:

- *Orthonormal matrices*. Generation and application of orthonormal matrices.
- *Unitary matrices*. Generation and application of unitary matrices.

3.3.1 Orthonormal matrices

List of functions for orthonormal matrices

- *hipsolverDn<type>orgbr_bufferSize()*
- *hipsolverDn<type>orgbr()*
- *hipsolverDn<type>orgqr_bufferSize()*
- *hipsolverDn<type>orgqr()*
- *hipsolverDn<type>orgtr_bufferSize()*
- *hipsolverDn<type>orgtr()*
- *hipsolverDn<type>ormqr_bufferSize()*
- *hipsolverDn<type>ormqr()*
- *hipsolverDn<type>ormtr_bufferSize()*
- *hipsolverDn<type>ormtr()*

hipsolverDn<type>orgbr_bufferSize()

hipsolverStatus_t **hipsolverDnDorgbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, **const** double *A, int lda, **const** double *tau, int *lwork)

hipsolverStatus_t **hipsolverDnSorgbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, **const** float *A, int lda, **const** float *tau, int *lwork)

hipsolverDn<type>orgbr()

hipsolverStatus_t **hipsolverDnDorgbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, double *A, int lda, **const** double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSorgbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, float *A, int lda, **const** float *tau, float *work, int lwork, int *devInfo)

hipsolverDn<type>orgqr_bufferSize()

hipsolverStatus_t **hipsolverDnDorgqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, **const** double *A, int lda, **const** double *tau, int *lwork)

hipsolverStatus_t **hipsolverDnSorgqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, **const** float *A, int lda, **const** float *tau, int *lwork)

hipsolverDn<type>orgqr()

hipsolverStatus_t **hipsolverDnDorgqr** (*hipsolverHandle_t* handle, int m, int n, int k, double *A, int lda, **const** double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSorgqr** (*hipsolverHandle_t* handle, int m, int n, int k, float *A, int lda, **const** float *tau, float *work, int lwork, int *devInfo)

hipsolverDn<type>orgtr_bufferSize()

hipsolverStatus_t **hipsolverDnDorgtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *tau, int *lwork)

hipsolverStatus_t **hipsolverDnSorgtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *tau, int *lwork)

hipsolverDn<type>orgtr()

hipsolverStatus_t **hipsolverDnDorgtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, **const** double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSorgtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, **const** float *tau, float *work, int lwork, int *devInfo)

hipsolverDn<type>ormqr_bufferSize()

hipsolverStatus_t **hipsolverDnDormqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** double *A, int lda, **const** double *tau, **const** double *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverDnSormqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** float *A, int lda, **const** float *tau, **const** float *C, int ldc, int *lwork)

hipsolverDn<type>ormqr()

hipsolverStatus_t **hipsolverDnDormqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** double *A, int lda, **const** double *tau, double *C, int ldc, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSormqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** float *A, int lda, **const** float *tau, float *C, int ldc, float *work, int lwork, int *devInfo)

hipsolverDn<type>ormtr_bufferSize()

hipsolverStatus_t **hipsolverDnDormtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, **const** double *A, int lda, **const** double *tau, **const** double *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverDnSormtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, **const** float *A, int lda, **const** float *tau, **const** float *C, int ldc, int *lwork)

hipsolverDn<type>ormtr()

hipsolverStatus_t **hipsolverDnDormtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, double *A, int lda, double *tau, double *C, int ldc, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSormtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, float *A, int lda, float *tau, float *C, int ldc, float *work, int lwork, int *devInfo)

3.3.2 Unitary matrices**List of functions for unitary matrices**

- *hipsolverDn<type>ungbr_bufferSize()*
- *hipsolverDn<type>ungbr()*
- *hipsolverDn<type>ungqr_bufferSize()*
- *hipsolverDn<type>ungqr()*
- *hipsolverDn<type>ungtr_bufferSize()*
- *hipsolverDn<type>ungtr()*
- *hipsolverDn<type>unmqr_bufferSize()*
- *hipsolverDn<type>unmqr()*
- *hipsolverDn<type>unmtr_bufferSize()*
- *hipsolverDn<type>unmtr()*

hipsolverDn<type>ungbr_bufferSize()

hipsolverStatus_t **hipsolverDnZungbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverDnCungbr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, int *lwork)

hipsolverDn<type>ungbr()

hipsolverStatus_t **hipsolverDnZungbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCungbr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, int m, int n, int k, hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>ungqr_bufferSize()

hipsolverStatus_t **hipsolverDnZungqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverDnCungqr_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int k, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, int *lwork)

hipsolverDn<type>ungqr()

hipsolverStatus_t **hipsolverDnZungqr** (*hipsolverHandle_t* handle, int m, int n, int k, hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCungqr** (*hipsolverHandle_t* handle, int m, int n, int k, hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>ungtr_bufferSize()

hipsolverStatus_t **hipsolverDnZungtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverDnCungtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, int *lwork)

hipsolverDn<type>ungtr()

hipsolverStatus_t **hipsolverDnZungtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCungtr** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>unmqr_bufferSize()

hipsolverStatus_t **hipsolverDnZunmqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, **const** hipDoubleComplex *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverDnCunmqr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, **const** hipFloatComplex *C, int ldc, int *lwork)

hipsolverDn<type>unmqr()

hipsolverStatus_t **hipsolverDnZunmqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCunmqr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverOperation_t* trans, int m, int n, int k, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, hipFloatComplex *C, int ldc, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>unmtr_bufferSize()

hipsolverStatus_t **hipsolverDnZunmtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *tau, **const** hipDoubleComplex *C, int ldc, int *lwork)

hipsolverStatus_t **hipsolverDnCunmtr_bufferSize** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *tau, **const** hipFloatComplex *C, int ldc, int *lwork)

hipsolverDn<type>unmtr()

hipsolverStatus_t **hipsolverDnZunmtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *C, int ldc, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCunmtr** (*hipsolverHandle_t* handle, *hipsolverSideMode_t* side, *hipsolverFillMode_t* uplo, *hipsolverOperation_t* trans, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *C, int ldc, hipFloatComplex *work, int lwork, int *devInfo)

3.4 LAPACK Functions

LAPACK routines solve complex Numerical Linear Algebra problems. These functions are organized in the following categories:

- *Triangular factorizations.* Based on Gaussian elimination.
- *Orthogonal factorizations.* Based on Householder reflections.
- *Problem and matrix reductions.* Transformation of matrices and problems into equivalent forms.
- *Linear-systems solvers.* Based on triangular factorizations.
- *Least-squares solvers.* Based on orthogonal factorizations.
- *Symmetric eigensolvers.* Eigenproblems for symmetric matrices.
- *Singular value decomposition.* Singular values and related problems for general matrices.

3.4.1 Triangular factorizations

List of triangular factorizations

- `hipsolverDn<type>potrf_bufferSize()`
- `hipsolverDn<type>potrf()`
- `hipsolverDn<type>potrfBatched()`
- `hipsolverDn<type>getrf_bufferSize()`
- `hipsolverDn<type>getrf()`
- `hipsolverDn<type>sytrf_bufferSize()`
- `hipsolverDn<type>sytrf()`

`hipsolverDn<type>potrf_bufferSize()`

`hipsolverStatus_t hipsolverDnZpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipDoubleComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnCpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipFloatComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnDpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnSpotrf_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `int *lwork`)

hipsolverDn<type>potrf()

hipsolverStatus_t **hipsolverDnZpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnDpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSpotrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *work, int lwork, int *devInfo)

hipsolverDn<type>potrfBatched()

hipsolverStatus_t **hipsolverDnZpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A[], int lda, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnCpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A[], int lda, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnDpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A[], int lda, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnSpotrfBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A[], int lda, int *devInfo, int batch_count)

hipsolverDn<type>getrf_bufferSize()

hipsolverStatus_t **hipsolverDnZgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnCgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnDgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnSgetrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, int *lwork)

hipsolverDn<type>getrf()

hipsolverStatus_t **hipsolverDnZgetrf** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *work, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverDnCgetrf** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *work, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverDnDgetrf** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, double *work, int *devI piv, int *devInfo)

hipsolverStatus_t **hipsolverDnSgetrf** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, float *work, int *devI piv, int *devInfo)

hipsolverDn<type>sytrf_bufferSize()

hipsolverStatus_t **hipsolverDnZsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnCsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnDsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnSsytrf_bufferSize** (*hipsolverHandle_t* handle, int n, float *A, int lda, int *lwork)

hipsolverDn<type>sytrf()

hipsolverStatus_t **hipsolverDnZsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, int *ipiv, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, int *ipiv, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnDsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, int *ipiv, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsytrf** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, int *ipiv, float *work, int lwork, int *devInfo)

3.4.2 Orthogonal factorizations

List of orthogonal factorizations

- *hipsolverDn<type>geqrf_bufferSize()*
- *hipsolverDn<type>geqrf()*

hipsolverDn<type>geqrf_bufferSize()

hipsolverStatus_t **hipsolverDnZgeqrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnCgeqrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnDgeqrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, int *lwork)

hipsolverStatus_t **hipsolverDnSgeqrf_bufferSize** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, int *lwork)

hipsolverDn<type>geqrf()

hipsolverStatus_t **hipsolverDnZgeqrf** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCgeqrf** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnDgeqrf** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSgeqrf** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, float *tau, float *work, int lwork, int *devInfo)

3.4.3 Problem and matrix reductions**List of reductions**

- *hipsolverDn<type>gebrd_bufferSize()*
- *hipsolverDn<type>gebrd()*
- *hipsolverDn<type>sytrd_bufferSize()*
- *hipsolverDn<type>hetrd_bufferSize()*
- *hipsolverDn<type>sytrd()*
- *hipsolverDn<type>hetrd()*

hipsolverDn<type>gebrd_bufferSize()

hipsolverStatus_t **hipsolverDnZgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnCgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnDgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnSgebrd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverDn<type>gebrd()

hipsolverStatus_t **hipsolverDnZgebrd** (*hipsolverHandle_t* handle, int m, int n, hipDoubleComplex *A, int lda, double *D, double *E, hipDoubleComplex *tauq, hipDoubleComplex *taup, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCgebrd** (*hipsolverHandle_t* handle, int m, int n, hipFloatComplex *A, int lda, float *D, float *E, hipFloatComplex *tauq, hipFloatComplex *taup, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnDgebrd** (*hipsolverHandle_t* handle, int m, int n, double *A, int lda, double *D, double *E, double *tauq, double *taup, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSgebrd** (*hipsolverHandle_t* handle, int m, int n, float *A, int lda, float *D, float *E, float *tauq, float *taup, float *work, int lwork, int *devInfo)

hipsolverDn<type>sytrd_bufferSize()

hipsolverStatus_t **hipsolverDnDsytrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, const double *A, int lda, const double *D, const double *E, const double *tau, int *lwork)

hipsolverStatus_t **hipsolverDnSsytrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, const float *A, int lda, const float *D, const float *E, const float *tau, int *lwork)

hipsolverDn<type>hetrd_bufferSize()

hipsolverStatus_t **hipsolverDnZhetrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, const hipDoubleComplex *A, int lda, const double *D, const double *E, const hipDoubleComplex *tau, int *lwork)

hipsolverStatus_t **hipsolverDnChetrd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, const hipFloatComplex *A, int lda, const float *D, const float *E, const hipFloatComplex *tau, int *lwork)

hipsolverDn<type>sytrd()

hipsolverStatus_t **hipsolverDnDsytrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *D, double *E, double *tau, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsytrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *D, float *E, float *tau, float *work, int lwork, int *devInfo)

hipsolverDn<type>hetrd()

hipsolverStatus_t **hipsolverDnZhetrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *D, double *E, hipDoubleComplex *tau, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnChetrd** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *D, float *E, hipFloatComplex *tau, hipFloatComplex *work, int lwork, int *devInfo)

3.4.4 Linear-systems solvers

List of linear solvers

- `hipsolverDn<type>potri_bufferSize()`
- `hipsolverDn<type>potri()`
- `hipsolverDn<type>potrs()`
- `hipsolverDn<type>potrsBatched()`
- `hipsolverDn<type>getrs()`
- `hipsolverDn<type><type>gesv_bufferSize()`
- `hipsolverDn<type><type>gesv()`

`hipsolverDn<type>potri_bufferSize()`

`hipsolverStatus_t hipsolverDnZpotri_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipDoubleComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnCpotri_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipFloatComplex *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnDpotri_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `int *lwork`)

`hipsolverStatus_t hipsolverDnSpotri_bufferSize` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `int *lwork`)

`hipsolverDn<type>potri()`

`hipsolverStatus_t hipsolverDnZpotri` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipDoubleComplex *A`, `int lda`, `hipDoubleComplex *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverDnCpotri` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `hipFloatComplex *A`, `int lda`, `hipFloatComplex *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverDnDpotri` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `double *A`, `int lda`, `double *work`, `int lwork`, `int *devInfo`)

`hipsolverStatus_t hipsolverDnSpotri` (`hipsolverHandle_t handle`, `hipsolverFillMode_t uplo`, `int n`, `float *A`, `int lda`, `float *work`, `int lwork`, `int *devInfo`)

hipsolverDn<type>potrs()

hipsolverStatus_t **hipsolverDnZpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, **const** hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnCpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, **const** hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnDpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, **const** double *A, int lda, double *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnSpotrs** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, **const** float *A, int lda, float *B, int ldb, int *devInfo)

hipsolverDn<type>potrsBatched()

hipsolverStatus_t **hipsolverDnZpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipDoubleComplex *A[], int lda, hipDoubleComplex *B[], int ldb, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnCpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, hipFloatComplex *A[], int lda, hipFloatComplex *B[], int ldb, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnDpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, double *A[], int lda, double *B[], int ldb, int *devInfo, int batch_count)

hipsolverStatus_t **hipsolverDnSpotrsBatched** (*hipsolverHandle_t* handle, *hipsolverFillMode_t* uplo, int n, int nrhs, float *A[], int lda, float *B[], int ldb, int *devInfo, int batch_count)

hipsolverDn<type>getrs()

hipsolverStatus_t **hipsolverDnZgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, **const** hipDoubleComplex *A, int lda, **const** int *devI piv, hipDoubleComplex *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnCgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, **const** hipFloatComplex *A, int lda, **const** int *devI piv, hipFloatComplex *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnDgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, **const** double *A, int lda, **const** int *devI piv, double *B, int ldb, int *devInfo)

hipsolverStatus_t **hipsolverDnSgetrs** (*hipsolverHandle_t* handle, *hipsolverOperation_t* trans, int n, int nrhs, **const** float *A, int lda, **const** int *devI piv, float *B, int ldb, int *devInfo)

hipsolverDn<type><type>gesv_bufferSize()

hipsolverStatus_t **hipsolverDnZZgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, hipDoubleComplex *A, int lda, int *devI piv, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnCCgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, hipFloatComplex *A, int lda, int *devI piv, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnDDgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, double *A, int lda, int *devI piv, double *B, int ldb, double *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnSSgesv_bufferSize** (*hipsolverHandle_t* handle, int n, int nrhs, float *A, int lda, int *devI piv, float *B, int ldb, float *X, int ldx, void *work, size_t *lwork)

hipsolverDn<type><type>gesv()

hipsolverStatus_t **hipsolverDnZZgesv** (*hipsolverHandle_t* handle, int n, int nrhs, hipDoubleComplex *A, int lda, int *devI piv, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverDnCCgesv** (*hipsolverHandle_t* handle, int n, int nrhs, hipFloatComplex *A, int lda, int *devI piv, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverDnDDgesv** (*hipsolverHandle_t* handle, int n, int nrhs, double *A, int lda, int *devI piv, double *B, int ldb, double *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

hipsolverStatus_t **hipsolverDnSSgesv** (*hipsolverHandle_t* handle, int n, int nrhs, float *A, int lda, int *devI piv, float *B, int ldb, float *X, int ldx, void *work, size_t lwork, int *niters, int *devInfo)

3.4.5 Least-squares solvers**List of least-squares solvers**

- *hipsolverDn<type><type>gels_bufferSize()*
- *hipsolverDn<type><type>gels()*

hipsolverDn<type><type>gels_bufferSize()

hipsolverStatus_t **hipsolverDnZZgels_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int nrhs, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnCCgels_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int nrhs, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnDDgels_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int nrhs, double *A, int lda, double *B, int ldb, double *X, int ldx, void *work, size_t *lwork)

hipsolverStatus_t **hipsolverDnSSgels_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int nrhs, float *A, int lda, float *B, int ldb, float *X, int ldx, void *work, size_t *lwork)

hipsolverDn<type><type>gels()

hipsolverStatus_t **hipsolverDnZZgels** (*hipsolverHandle_t* handle, int m, int n, int nrhs, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, hipDoubleComplex *X, int ldx, void *work, size_t lwork, int *nitters, int *devInfo)

hipsolverStatus_t **hipsolverDnCCgels** (*hipsolverHandle_t* handle, int m, int n, int nrhs, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, hipFloatComplex *X, int ldx, void *work, size_t lwork, int *nitters, int *devInfo)

hipsolverStatus_t **hipsolverDnDDgels** (*hipsolverHandle_t* handle, int m, int n, int nrhs, double *A, int lda, double *B, int ldb, double *X, int ldx, void *work, size_t lwork, int *nitters, int *devInfo)

hipsolverStatus_t **hipsolverDnSSgels** (*hipsolverHandle_t* handle, int m, int n, int nrhs, float *A, int lda, float *B, int ldb, float *X, int ldx, void *work, size_t lwork, int *nitters, int *devInfo)

3.4.6 Symmetric eigensolvers**List of symmetric eigensolvers**

- *hipsolverDn<type>syevd_bufferSize()*
- *hipsolverDn<type>heevd_bufferSize()*
- *hipsolverDn<type>syevd()*
- *hipsolverDn<type>heevd()*
- *hipsolverDn<type>sygvd_bufferSize()*
- *hipsolverDn<type>hegvd_bufferSize()*
- *hipsolverDn<type>sygvd()*
- *hipsolverDn<type>hegvd()*

hipsolverDn<type>syevd_bufferSize()

hipsolverStatus_t **hipsolverDnDsyevd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnSsyevd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *W, int *lwork)

hipsolverDn<type>heevd_bufferSize()

hipsolverStatus_t **hipsolverDnZheevd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnCheevd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** float *W, int *lwork)

hipsolverDn<type>syevd()

hipsolverStatus_t **hipsolverDnDsyevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsyevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, float *work, int lwork, int *devInfo)

hipsolverDn<type>heevd()

hipsolverStatus_t **hipsolverDnZheevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCheevd** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>sygvd_bufferSize()

hipsolverStatus_t **hipsolverDnDsygvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *B, int ldb, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnSsygvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *B, int ldb, **const** float *W, int *lwork)

hipsolverDn<type>hegvd_bufferSize()

hipsolverStatus_t **hipsolverDnZhegvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *B, int ldb, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnChegvd_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *B, int ldb, **const** float *W, int *lwork)

hipsolverDn<type>sygvd()

hipsolverStatus_t **hipsolverDnDsygvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsygvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, float *work, int lwork, int *devInfo)

hipsolverDn<type>hegvd()

hipsolverStatus_t **hipsolverDnZhegvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnChegvd** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, hipFloatComplex *work, int lwork, int *devInfo)

3.4.7 Singular value decomposition**List of SVD related functions**

- *hipsolverDn<type>gesvd_bufferSize()*
- *hipsolverDn<type>gesvd()*

hipsolverDn<type>gesvd_bufferSize()

hipsolverStatus_t **hipsolverDnZgesvd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnCgesvd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnDgesvd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverStatus_t **hipsolverDnSgesvd_bufferSize** (*hipsolverHandle_t* handle, int m, int n, int *lwork)

hipsolverDn<type>gesvd()

hipsolverStatus_t **hipsolverDnZgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, double *rwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, float *rwork, int *devInfo)

hipsolverStatus_t **hipsolverDnDgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, double *rwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSgesvd** (*hipsolverHandle_t* handle, signed char jobu, signed char jobv, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, float *rwork, int *devInfo)

3.5 LAPACK-like Functions

Other Lapack-like routines provided by hipSOLVER. These are divided into the following subcategories:

- *Symmetric eigensolvers*. Eigenproblems for symmetric matrices.
- *Singular value decomposition*. Singular values and related problems for general matrices.

3.5.1 Symmetric eigensolvers

List of Lapack-like symmetric eigensolvers

- *hipsolverDn<type>syevdx_bufferSize()*
- *hipsolverDn<type>heevdx_bufferSize()*
- *hipsolverDn<type>syevdx()*
- *hipsolverDn<type>heevdx()*
- *hipsolverDn<type>syevj_bufferSize()*
- *hipsolverDn<type>heevj_bufferSize()*

- `hipsolverDn<type>syevjBatched_bufferSize()`
- `hipsolverDn<type>heevjBatched_bufferSize()`
- `hipsolverDn<type>syevj()`
- `hipsolverDn<type>heevj()`
- `hipsolverDn<type>syevjBatched()`
- `hipsolverDn<type>heevjBatched()`
- `hipsolverDn<type>sygvdx_bufferSize()`
- `hipsolverDn<type>hegvdx_bufferSize()`
- `hipsolverDn<type>sygvdx()`
- `hipsolverDn<type>hegvdx()`
- `hipsolverDn<type>sygvj_bufferSize()`
- `hipsolverDn<type>hegvj_bufferSize()`
- `hipsolverDn<type>sygvj()`
- `hipsolverDn<type>hegvj()`

`hipsolverDn<type>syevdx_bufferSize()`

`hipsolverStatus_t hipsolverDnDsyevdx_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverEigRange_t range`, `hipsolverFillMode_t uplo`, `int n`, `const double *A`, `int lda`, `double vl`, `double vu`, `int il`, `int iu`, `int *nev`, `const double *W`, `int *lwork`)

`hipsolverStatus_t hipsolverDnSsyevdx_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverEigRange_t range`, `hipsolverFillMode_t uplo`, `int n`, `const float *A`, `int lda`, `float vl`, `float vu`, `int il`, `int iu`, `int *nev`, `const float *W`, `int *lwork`)

`hipsolverDn<type>heevdx_bufferSize()`

`hipsolverStatus_t hipsolverDnZheevdx_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverEigRange_t range`, `hipsolverFillMode_t uplo`, `int n`, `const hipDoubleComplex *A`, `int lda`, `double vl`, `double vu`, `int il`, `int iu`, `int *nev`, `const double *W`, `int *lwork`)

`hipsolverStatus_t hipsolverDnCHeevdx_bufferSize` (`hipsolverHandle_t handle`, `hipsolverEigMode_t jobz`, `hipsolverEigRange_t range`, `hipsolverFillMode_t uplo`, `int n`, `const hipFloatComplex *A`, `int lda`, `float vl`, `float vu`, `int il`, `int iu`, `int *nev`, `const float *W`, `int *lwork`)

hipsolverDn<type>syevdx()

hipsolverStatus_t **hipsolverDnDsyevdx** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double vl, double vu, int il, int iu, int *nev, double *W, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsyevdx** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float vl, float vu, int il, int iu, int *nev, float *W, float *work, int lwork, int *devInfo)

hipsolverDn<type>heevdx()

hipsolverStatus_t **hipsolverDnZheevdx** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double vl, double vu, int il, int iu, int *nev, double *W, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnCheevdx** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float vl, float vu, int il, int iu, int *nev, float *W, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>syevj_bufferSize()

hipsolverStatus_t **hipsolverDnDsyevj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, const double *A, int lda, const double *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverDnSsyevj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, const float *A, int lda, const float *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverDn<type>heevj_bufferSize()

hipsolverStatus_t **hipsolverDnZheevj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, const hipDoubleComplex *A, int lda, const double *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverDnCheevj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, const hipFloatComplex *A, int lda, const float *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverDn<type>syevjBatched_bufferSize()

hipsolverStatus_t **hipsolverDnDsyevjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnSsyevjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverDn<type>heevjBatched_bufferSize()

hipsolverStatus_t **hipsolverDnZheevjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** double *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnCheevjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** float *W, int *lwork, *hipsolverSyevjInfo_t* params, int batch_count)

hipsolverDn<type>syevj()

hipsolverStatus_t **hipsolverDnDsyevj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, double *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverDnSsyevj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, float *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverDn<type>heevj()

hipsolverStatus_t **hipsolverDnZheevj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverDnCheevj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyevjInfo_t* params)

hipsolverDn<type>syevjBatched()

hipsolverStatus_t **hipsolverDnDsyejvBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *W, double *work, int lwork, int *devInfo, *hipsolverSyejvInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnSsyevjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *W, float *work, int lwork, int *devInfo, *hipsolverSyejvInfo_t* params, int batch_count)

hipsolverDn<type>heevjBatched()

hipsolverStatus_t **hipsolverDnZheevjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyejvInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnCheevjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyejvInfo_t* params, int batch_count)

hipsolverDn<type>sygvdx_bufferSize()

hipsolverStatus_t **hipsolverDnDsygvdx_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *B, int ldb, double vl, double vu, int il, int iu, int *nev, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnSsygvdx_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *B, int ldb, float vl, float vu, int il, int iu, int *nev, **const** float *W, int *lwork)

hipsolverDn<type>hegvdx_bufferSize()

hipsolverStatus_t **hipsolverDnZhegvdx_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *B, int ldb, double vl, double vu, int il, int iu, int *nev, **const** double *W, int *lwork)

hipsolverStatus_t **hipsolverDnChegvdx_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *B, int ldb, float vl, float vu, int il, int iu, int *nev, **const** float *W, int *lwork)

hipsolverDn<type>sygvdx()

hipsolverStatus_t **hipsolverDnDsygvdx** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double vl, double vu, int il, int iu, int *nev, double *W, double *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnSsygvdx** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float vl, float vu, int il, int iu, int *nev, float *W, float *work, int lwork, int *devInfo)

hipsolverDn<type>hegvdx()

hipsolverStatus_t **hipsolverDnZhegvdx** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double vl, double vu, int il, int iu, int *nev, double *W, hipDoubleComplex *work, int lwork, int *devInfo)

hipsolverStatus_t **hipsolverDnChegvdx** (*hipsolverHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverEigRange_t* range, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float vl, float vu, int il, int iu, int *nev, float *W, hipFloatComplex *work, int lwork, int *devInfo)

hipsolverDn<type>sygvj_bufferSize()

hipsolverStatus_t **hipsolverDnDsygvj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** double *A, int lda, **const** double *B, int ldb, **const** double *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverStatus_t **hipsolverDnSsygvj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** float *A, int lda, **const** float *B, int ldb, **const** float *W, int *lwork, *hipsolverSyevjInfo_t* params)

hipsolverDn<type>hegvj_bufferSize()

hipsolverStatus_t **hipsolverDnZhegvj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipDoubleComplex *A, int lda, **const** hipDoubleComplex *B, int ldb, **const** double *W, int *lwork, *hipsolverSyejInfo_t* params)

hipsolverStatus_t **hipsolverDnChegvj_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, **const** hipFloatComplex *A, int lda, **const** hipFloatComplex *B, int ldb, **const** float *W, int *lwork, *hipsolverSyejInfo_t* params)

hipsolverDn<type>sygvj()

hipsolverStatus_t **hipsolverDnDsygvj** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, double *A, int lda, double *B, int ldb, double *W, double *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

hipsolverStatus_t **hipsolverDnSsygvj** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, float *A, int lda, float *B, int ldb, float *W, float *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

hipsolverDn<type>hegvj()

hipsolverStatus_t **hipsolverDnZhegvj** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipDoubleComplex *A, int lda, hipDoubleComplex *B, int ldb, double *W, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

hipsolverStatus_t **hipsolverDnChegvj** (*hipsolverDnHandle_t* handle, *hipsolverEigType_t* itype, *hipsolverEigMode_t* jobz, *hipsolverFillMode_t* uplo, int n, hipFloatComplex *A, int lda, hipFloatComplex *B, int ldb, float *W, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverSyejInfo_t* params)

3.5.2 Singular value decomposition**List of Lapack-like SVD related functions**

- *hipsolverDn<type>gesvdj_bufferSize()*
- *hipsolverDn<type>gesvdjBatched_bufferSize()*
- *hipsolverDn<type>gesvdj()*
- *hipsolverDn<type>gesvdjBatched()*

- `hipsolverDn<type>gesvdaStridedBatched_bufferSize()`
- `hipsolverDn<type>gesvdaStridedBatched()`

`hipsolverDn<type>gesvdj_bufferSize()`

`hipsolverStatus_t hipsolverDnZgesvdj_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int econ, int m, int n, **const** hipDoubleComplex *A, int lda, **const** double *S, **const** hipDoubleComplex *U, int ldu, **const** hipDoubleComplex *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params)

`hipsolverStatus_t hipsolverDnCgesvdj_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int econ, int m, int n, **const** hipFloatComplex *A, int lda, **const** float *S, **const** hipFloatComplex *U, int ldu, **const** hipFloatComplex *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params)

`hipsolverStatus_t hipsolverDnDgesvdj_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int econ, int m, int n, **const** double *A, int lda, **const** double *S, **const** double *U, int ldu, **const** double *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params)

`hipsolverStatus_t hipsolverDnSgesvdj_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int econ, int m, int n, **const** float *A, int lda, **const** float *S, **const** float *U, int ldu, **const** float *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params)

`hipsolverDn<type>gesvdjBatched_bufferSize()`

`hipsolverStatus_t hipsolverDnZgesvdjBatched_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int m, int n, **const** hipDoubleComplex *A, int lda, **const** double *S, **const** hipDoubleComplex *U, int ldu, **const** hipDoubleComplex *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params, int batch_count)

`hipsolverStatus_t hipsolverDnCgesvdjBatched_bufferSize` (`hipsolverDnHandle_t` handle, `hipsolverEigMode_t` jobz, int m, int n, **const** hipFloatComplex *A, int lda, **const** float *S, **const** hipFloatComplex *U, int ldu, **const** hipFloatComplex *V, int ldv, int *lwork, `hipsolverGesvdjInfo_t` params, int batch_count)

hipsolverStatus_t **hipsolverDnDgesvdjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, **const** double *A, int lda, **const** double *S, **const** double *U, int ldu, **const** double *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnSgesvdjBatched_bufferSize** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, **const** float *A, int lda, **const** float *S, **const** float *U, int ldu, **const** float *V, int ldv, int *lwork, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverDn<type>gesvdj()

hipsolverStatus_t **hipsolverDnZgesvdj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverDnCgesvdj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverDnDgesvdj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverStatus_t **hipsolverDnSgesvdj** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int econ, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params)

hipsolverDn<type>gesvdjBatched()

hipsolverStatus_t **hipsolverDnZgesvdjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, hipDoubleComplex *A, int lda, double *S, hipDoubleComplex *U, int ldu, hipDoubleComplex *V, int ldv, hipDoubleComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnCgesvdjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, hipFloatComplex *A, int lda, float *S, hipFloatComplex *U, int ldu, hipFloatComplex *V, int ldv, hipFloatComplex *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnDgesvdjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, double *A, int lda, double *S, double *U, int ldu, double *V, int ldv, double *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverStatus_t **hipsolverDnSgesvdjBatched** (*hipsolverDnHandle_t* handle, *hipsolverEigMode_t* jobz, int m, int n, float *A, int lda, float *S, float *U, int ldu, float *V, int ldv, float *work, int lwork, int *devInfo, *hipsolverGesvdjInfo_t* params, int batch_count)

hipsolverDn<type>gesvdaStridedBatched_bufferSize()

hipsolverStatus_t **hipsolverDnZgesvdaStridedBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** hipDoubleComplex *A, int lda, long long int strideA, **const** double *S, long long int strideS, **const** hipDoubleComplex *U, int ldu, long long int strideU, **const** hipDoubleComplex *V, int ldv, long long int strideV, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverDnCgesvdaStridedBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** hipFloatComplex *A, int lda, long long int strideA, **const** float *S, long long int strideS, **const** hipFloatComplex *U, int ldu, long long int strideU, **const** hipFloatComplex *V, int ldv, long long int strideV, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverDnDgesvdaStridedBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** double *A, int lda, long long int strideA, **const** double *S, long long int strideS, **const** double *U, int ldu, long long int strideU, **const** double *V, int ldv, long long int strideV, int *lwork, int batch_count)

hipsolverStatus_t **hipsolverDnSgesvdaStridedBatched_bufferSize** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** float *A, int lda, long long int strideA, **const** float *S, long long int strideS, **const** float *U, int ldu, long long int strideU, **const** float *V, int ldv, long long int strideV, int *lwork, int batch_count)

hipsolverDn<type>gesvdaStridedBatched()

hipsolverStatus_t **hipsolverDnZgesvdaStridedBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** hipDoubleComplex *A, int lda, long long int strideA, double *S, long long int strideS, hipDoubleComplex *U, int ldu, long long int strideU, hipDoubleComplex *V, int ldv, long long int strideV, hipDoubleComplex *work, int lwork, int *devInfo, double *hRnormF, int batch_count)

hipsolverStatus_t **hipsolverDnCgesvdaStridedBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** hipFloatComplex *A, int lda, long long int strideA, float *S, long long int strideS, hipFloatComplex *U, int ldu, long long int strideU, hipFloatComplex *V, int ldv, long long int strideV, hipFloatComplex *work, int lwork, int *devInfo, double *hRnormF, int batch_count)

hipsolverStatus_t **hipsolverDnDgesvdaStridedBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** double *A, int lda, long long int strideA, double *S, long long int strideS, double *U, int ldu, long long int strideU, double *V, int ldv, long long int strideV, double *work, int lwork, int *devInfo, double *hRnormF, int batch_count)

hipsolverStatus_t **hipsolverDnSgesvdaStridedBatched** (*hipsolverHandle_t* handle, *hipsolverEigMode_t* jobz, int rank, int m, int n, **const** float *A, int lda, long long int strideA, float *S, long long int strideS, float *U, int ldu, long long int strideU, float *V, int ldv, long long int strideV, float *work, int lwork, int *devInfo, double *hRnormF, int batch_count)

H

- hipsolverCCgels (*C function*), 33
- hipsolverCCgels_bufferSize (*C function*), 33
- hipsolverCCgesv (*C function*), 32
- hipsolverCCgesv_bufferSize (*C function*), 32
- hipsolverCgebrd (*C function*), 28
- hipsolverCgebrd_bufferSize (*C function*), 28
- hipsolverCgeqrf (*C function*), 27
- hipsolverCgeqrf_bufferSize (*C function*), 27
- hipsolverCgesvd (*C function*), 36
- hipsolverCgesvd_bufferSize (*C function*), 36
- hipsolverCgesvdj (*C function*), 42
- hipsolverCgesvdj_bufferSize (*C function*), 41
- hipsolverCgesvdjBatched (*C function*), 42
- hipsolverCgesvdjBatched_bufferSize (*C function*), 41
- hipsolverCgetrf (*C function*), 26
- hipsolverCgetrf_bufferSize (*C function*), 26
- hipsolverCgetrs (*C function*), 32
- hipsolverCgetrs_bufferSize (*C function*), 31
- hipsolverCheevd (*C function*), 35
- hipsolverCheevd_bufferSize (*C function*), 34
- hipsolverCheevj (*C function*), 39
- hipsolverCheevj_bufferSize (*C function*), 38
- hipsolverCheevjBatched (*C function*), 39
- hipsolverCheevjBatched_bufferSize (*C function*), 38
- hipsolverChegvd (*C function*), 36
- hipsolverChegvd_bufferSize (*C function*), 35
- hipsolverChegvj (*C function*), 40
- hipsolverChegvj_bufferSize (*C function*), 40
- hipsolverChetrd (*C function*), 29
- hipsolverChetrd_bufferSize (*C function*), 28
- hipsolverCpotrf (*C function*), 25
- hipsolverCpotrf_bufferSize (*C function*), 24
- hipsolverCpotrfBatched (*C function*), 25
- hipsolverCpotrfBatched_bufferSize (*C function*), 25
- hipsolverCpotri (*C function*), 30
- hipsolverCpotri_bufferSize (*C function*), 29
- hipsolverCpotrs (*C function*), 31
- hipsolverCpotrs_bufferSize (*C function*), 30
- hipsolverCpotrsBatched (*C function*), 31
- hipsolverCpotrsBatched_bufferSize (*C function*), 30
- hipsolverCreate (*C function*), 16
- hipsolverCreateGesvdjInfo (*C function*), 17
- hipsolverCreateSyevjInfo (*C function*), 18
- hipsolverCsytrf (*C function*), 26
- hipsolverCsytrf_bufferSize (*C function*), 26
- hipsolverCungbr (*C function*), 22
- hipsolverCungbr_bufferSize (*C function*), 21
- hipsolverCungqr (*C function*), 22
- hipsolverCungqr_bufferSize (*C function*), 22
- hipsolverCungtr (*C function*), 22
- hipsolverCungtr_bufferSize (*C function*), 22
- hipsolverCunmqr (*C function*), 23
- hipsolverCunmqr_bufferSize (*C function*), 23
- hipsolverCunmtr (*C function*), 23
- hipsolverCunmtr_bufferSize (*C function*), 23
- hipsolverDDgels (*C function*), 33
- hipsolverDDgels_bufferSize (*C function*), 33
- hipsolverDDgesv (*C function*), 32
- hipsolverDDgesv_bufferSize (*C function*), 32
- hipsolverDestroy (*C function*), 16
- hipsolverDestroyGesvdjInfo (*C function*), 17
- hipsolverDestroySyevjInfo (*C function*), 18
- hipsolverDgebrd (*C function*), 28
- hipsolverDgebrd_bufferSize (*C function*), 28
- hipsolverDgeqrf (*C function*), 27
- hipsolverDgeqrf_bufferSize (*C function*), 27
- hipsolverDgesvd (*C function*), 36
- hipsolverDgesvd_bufferSize (*C function*), 36
- hipsolverDgesvdj (*C function*), 42
- hipsolverDgesvdj_bufferSize (*C function*), 41
- hipsolverDgesvdjBatched (*C function*), 42
- hipsolverDgesvdjBatched_bufferSize (*C function*), 41
- hipsolverDgetrf (*C function*), 26
- hipsolverDgetrf_bufferSize (*C function*), 26
- hipsolverDgetrs (*C function*), 32
- hipsolverDgetrs_bufferSize (*C function*), 31
- hipsolverDnCCgels (*C function*), 62
- hipsolverDnCCgels_bufferSize (*C function*),

- 62
 hipsolverDnCCgesv (*C function*), 61
 hipsolverDnCCgesv_bufferSize (*C function*), 61
 hipsolverDnCgebrd (*C function*), 57
 hipsolverDnCgebrd_bufferSize (*C function*), 57
 hipsolverDnCgeqrf (*C function*), 57
 hipsolverDnCgeqrf_bufferSize (*C function*), 56
 hipsolverDnCgesvd (*C function*), 65
 hipsolverDnCgesvd_bufferSize (*C function*), 65
 hipsolverDnCgesvdaStridedBatched (*C function*), 75
 hipsolverDnCgesvdaStridedBatched_bufferSize (*C function*), 74
 hipsolverDnCgesvdj (*C function*), 73
 hipsolverDnCgesvdj_bufferSize (*C function*), 72
 hipsolverDnCgesvdjBatched (*C function*), 73
 hipsolverDnCgesvdjBatched_bufferSize (*C function*), 72
 hipsolverDnCgetrf (*C function*), 55
 hipsolverDnCgetrf_bufferSize (*C function*), 55
 hipsolverDnCgetrs (*C function*), 60
 hipsolverDnCheevd (*C function*), 63
 hipsolverDnCheevd_bufferSize (*C function*), 63
 hipsolverDnCheevdx (*C function*), 67
 hipsolverDnCheevdx_bufferSize (*C function*), 66
 hipsolverDnCheevj (*C function*), 68
 hipsolverDnCheevj_bufferSize (*C function*), 67
 hipsolverDnCheevjBatched (*C function*), 69
 hipsolverDnCheevjBatched_bufferSize (*C function*), 68
 hipsolverDnChegvd (*C function*), 64
 hipsolverDnChegvd_bufferSize (*C function*), 64
 hipsolverDnChegvdv (*C function*), 70
 hipsolverDnChegvdv_bufferSize (*C function*), 69
 hipsolverDnChegvj (*C function*), 71
 hipsolverDnChegvj_bufferSize (*C function*), 71
 hipsolverDnChetrd (*C function*), 58
 hipsolverDnChetrd_bufferSize (*C function*), 58
 hipsolverDnCpotrf (*C function*), 55
 hipsolverDnCpotrf_bufferSize (*C function*), 54
 hipsolverDnCpotrfBatched (*C function*), 55
 hipsolverDnCpotri (*C function*), 59
 hipsolverDnCpotri_bufferSize (*C function*), 59
 hipsolverDnCpotrs (*C function*), 60
 hipsolverDnCpotrsBatched (*C function*), 60
 hipsolverDnCreate (*C function*), 45
 hipsolverDnCreateGesvdjInfo (*C function*), 46
 hipsolverDnCreateSyevjInfo (*C function*), 47
 hipsolverDnCsytrf (*C function*), 56
 hipsolverDnCsytrf_bufferSize (*C function*), 56
 hipsolverDnCungbr (*C function*), 52
 hipsolverDnCungbr_bufferSize (*C function*), 51
 hipsolverDnCungqr (*C function*), 52
 hipsolverDnCungqr_bufferSize (*C function*), 52
 hipsolverDnCungtr (*C function*), 52
 hipsolverDnCungtr_bufferSize (*C function*), 52
 hipsolverDnCumqr (*C function*), 53
 hipsolverDnCumqr_bufferSize (*C function*), 53
 hipsolverDnCumtr (*C function*), 53
 hipsolverDnCumtr_bufferSize (*C function*), 53
 hipsolverDnDDgels (*C function*), 62
 hipsolverDnDDgels_bufferSize (*C function*), 62
 hipsolverDnDDgesv (*C function*), 61
 hipsolverDnDDgesv_bufferSize (*C function*), 61
 hipsolverDnDestroy (*C function*), 45
 hipsolverDnDestroyGesvdjInfo (*C function*), 46
 hipsolverDnDestroySyevjInfo (*C function*), 47
 hipsolverDnDgebrd (*C function*), 57
 hipsolverDnDgebrd_bufferSize (*C function*), 57
 hipsolverDnDgeqrf (*C function*), 57
 hipsolverDnDgeqrf_bufferSize (*C function*), 56
 hipsolverDnDgesvd (*C function*), 65
 hipsolverDnDgesvd_bufferSize (*C function*), 65
 hipsolverDnDgesvdaStridedBatched (*C function*), 75
 hipsolverDnDgesvdaStridedBatched_bufferSize (*C function*), 74
 hipsolverDnDgesvdj (*C function*), 73
 hipsolverDnDgesvdj_bufferSize (*C function*), 72
 hipsolverDnDgesvdjBatched (*C function*), 73

hipsolverDnDgesvdjBatched_bufferSize (C function), 72
 hipsolverDnDgetrf (C function), 55
 hipsolverDnDgetrf_bufferSize (C function), 55
 hipsolverDnDgetrs (C function), 60
 hipsolverDnDorgbr (C function), 49
 hipsolverDnDorgbr_bufferSize (C function), 49
 hipsolverDnDorgqr (C function), 49
 hipsolverDnDorgqr_bufferSize (C function), 49
 hipsolverDnDorgtr (C function), 50
 hipsolverDnDorgtr_bufferSize (C function), 49
 hipsolverDnDormqr (C function), 50
 hipsolverDnDormqr_bufferSize (C function), 50
 hipsolverDnDormtr (C function), 51
 hipsolverDnDormtr_bufferSize (C function), 50
 hipsolverDnDpotrf (C function), 55
 hipsolverDnDpotrf_bufferSize (C function), 54
 hipsolverDnDpotrfBatched (C function), 55
 hipsolverDnDpotri (C function), 59
 hipsolverDnDpotri_bufferSize (C function), 59
 hipsolverDnDpotrs (C function), 60
 hipsolverDnDpotrsBatched (C function), 60
 hipsolverDnDsyevd (C function), 63
 hipsolverDnDsyevd_bufferSize (C function), 63
 hipsolverDnDsyevdx (C function), 67
 hipsolverDnDsyevdx_bufferSize (C function), 66
 hipsolverDnDsyevj (C function), 68
 hipsolverDnDsyevj_bufferSize (C function), 67
 hipsolverDnDsyevjBatched (C function), 69
 hipsolverDnDsyevjBatched_bufferSize (C function), 68
 hipsolverDnDsygvd (C function), 64
 hipsolverDnDsygvd_bufferSize (C function), 63
 hipsolverDnDsygvdx (C function), 70
 hipsolverDnDsygvdx_bufferSize (C function), 69
 hipsolverDnDsygvj (C function), 71
 hipsolverDnDsygvj_bufferSize (C function), 70
 hipsolverDnDsytrd (C function), 58
 hipsolverDnDsytrd_bufferSize (C function), 58
 hipsolverDnDsytrf (C function), 56
 hipsolverDnDsytrf_bufferSize (C function), 56
 hipsolverDnGetStream (C function), 46
 hipsolverDnHandle_t (C type), 44
 hipsolverDnSetStream (C function), 45
 hipsolverDnSgebrd (C function), 58
 hipsolverDnSgebrd_bufferSize (C function), 57
 hipsolverDnSgeqrf (C function), 57
 hipsolverDnSgeqrf_bufferSize (C function), 56
 hipsolverDnSgesvd (C function), 65
 hipsolverDnSgesvd_bufferSize (C function), 65
 hipsolverDnSgesvdaStridedBatched (C function), 75
 hipsolverDnSgesvdaStridedBatched_bufferSize (C function), 74
 hipsolverDnSgesvdj (C function), 73
 hipsolverDnSgesvdj_bufferSize (C function), 72
 hipsolverDnSgesvdjBatched (C function), 74
 hipsolverDnSgesvdjBatched_bufferSize (C function), 73
 hipsolverDnSgetrf (C function), 55
 hipsolverDnSgetrf_bufferSize (C function), 55
 hipsolverDnSgetrs (C function), 60
 hipsolverDnSorgbr (C function), 49
 hipsolverDnSorgbr_bufferSize (C function), 49
 hipsolverDnSorgqr (C function), 49
 hipsolverDnSorgqr_bufferSize (C function), 49
 hipsolverDnSorgtr (C function), 50
 hipsolverDnSorgtr_bufferSize (C function), 49
 hipsolverDnSormqr (C function), 50
 hipsolverDnSormqr_bufferSize (C function), 50
 hipsolverDnSormtr (C function), 51
 hipsolverDnSormtr_bufferSize (C function), 50
 hipsolverDnSpotrf (C function), 55
 hipsolverDnSpotrf_bufferSize (C function), 54
 hipsolverDnSpotrfBatched (C function), 55
 hipsolverDnSpotri (C function), 59
 hipsolverDnSpotri_bufferSize (C function), 59
 hipsolverDnSpotrs (C function), 60
 hipsolverDnSpotrsBatched (C function), 60
 hipsolverDnSSgels (C function), 62

hipsolverDnSSgels_bufferSize (C function), 62

hipsolverDnSSgesv (C function), 61

hipsolverDnSSgesv_bufferSize (C function), 61

hipsolverDnSsyevd (C function), 63

hipsolverDnSsyevd_bufferSize (C function), 63

hipsolverDnSsyevdx (C function), 67

hipsolverDnSsyevdx_bufferSize (C function), 66

hipsolverDnSsyevj (C function), 68

hipsolverDnSsyevj_bufferSize (C function), 67

hipsolverDnSsyevjBatched (C function), 69

hipsolverDnSsyevjBatched_bufferSize (C function), 68

hipsolverDnSsygvd (C function), 64

hipsolverDnSsygvd_bufferSize (C function), 63

hipsolverDnSsygvdx (C function), 70

hipsolverDnSsygvdx_bufferSize (C function), 69

hipsolverDnSsygvj (C function), 71

hipsolverDnSsygvj_bufferSize (C function), 70

hipsolverDnSsytrd (C function), 58

hipsolverDnSsytrd_bufferSize (C function), 58

hipsolverDnSsytrf (C function), 56

hipsolverDnSsytrf_bufferSize (C function), 56

hipsolverDnXgesvdjGetResidual (C function), 47

hipsolverDnXgesvdjGetSweeps (C function), 47

hipsolverDnXgesvdjSetMaxSweeps (C function), 46

hipsolverDnXgesvdjSetSortEig (C function), 46

hipsolverDnXgesvdjSetTolerance (C function), 46

hipsolverDnXsyevjGetResidual (C function), 48

hipsolverDnXsyevjGetSweeps (C function), 48

hipsolverDnXsyevjSetMaxSweeps (C function), 47

hipsolverDnXsyevjSetSortEig (C function), 47

hipsolverDnXsyevjSetTolerance (C function), 48

hipsolverDnZgebrd (C function), 57

hipsolverDnZgebrd_bufferSize (C function), 57

hipsolverDnZgeqrf (C function), 57

hipsolverDnZgeqrf_bufferSize (C function), 56

hipsolverDnZgesvd (C function), 65

hipsolverDnZgesvd_bufferSize (C function), 65

hipsolverDnZgesvdaStridedBatched (C function), 75

hipsolverDnZgesvdaStridedBatched_bufferSize (C function), 74

hipsolverDnZgesvdj (C function), 73

hipsolverDnZgesvdj_bufferSize (C function), 72

hipsolverDnZgesvdjBatched (C function), 73

hipsolverDnZgesvdjBatched_bufferSize (C function), 72

hipsolverDnZgetrf (C function), 55

hipsolverDnZgetrf_bufferSize (C function), 55

hipsolverDnZgetrs (C function), 60

hipsolverDnZheevd (C function), 63

hipsolverDnZheevd_bufferSize (C function), 63

hipsolverDnZheevdx (C function), 67

hipsolverDnZheevdx_bufferSize (C function), 66

hipsolverDnZheevj (C function), 68

hipsolverDnZheevj_bufferSize (C function), 67

hipsolverDnZheevjBatched (C function), 69

hipsolverDnZheevjBatched_bufferSize (C function), 68

hipsolverDnZhegvd (C function), 64

hipsolverDnZhegvd_bufferSize (C function), 64

hipsolverDnZhegvdx (C function), 70

hipsolverDnZhegvdx_bufferSize (C function), 69

hipsolverDnZhegvj (C function), 71

hipsolverDnZhegvj_bufferSize (C function), 71

hipsolverDnZhetrd (C function), 58

hipsolverDnZhetrd_bufferSize (C function), 58

hipsolverDnZpotrf (C function), 55

hipsolverDnZpotrf_bufferSize (C function), 54

hipsolverDnZpotrfBatched (C function), 55

hipsolverDnZpotri (C function), 59

hipsolverDnZpotri_bufferSize (C function), 59

hipsolverDnZpotrs (C function), 60

hipsolverDnZpotrsBatched (C function), 60

hipsolverDnZsytrf (C function), 56

hipsolverDnZsytrf_bufferSize (C function), 56

- hipsolverDnZungbr (*C function*), 52
 hipsolverDnZungbr_bufferSize (*C function*), 51
 hipsolverDnZungqr (*C function*), 52
 hipsolverDnZungqr_bufferSize (*C function*), 52
 hipsolverDnZungtr (*C function*), 52
 hipsolverDnZungtr_bufferSize (*C function*), 52
 hipsolverDnZumqr (*C function*), 53
 hipsolverDnZumqr_bufferSize (*C function*), 53
 hipsolverDnZumtr (*C function*), 53
 hipsolverDnZumtr_bufferSize (*C function*), 53
 hipsolverDnZZgels (*C function*), 62
 hipsolverDnZZgels_bufferSize (*C function*), 62
 hipsolverDnZZgesv (*C function*), 61
 hipsolverDnZZgesv_bufferSize (*C function*), 61
 hipsolverDorgbr (*C function*), 19
 hipsolverDorgbr_bufferSize (*C function*), 19
 hipsolverDorgqr (*C function*), 19
 hipsolverDorgqr_bufferSize (*C function*), 19
 hipsolverDorgtr (*C function*), 20
 hipsolverDorgtr_bufferSize (*C function*), 20
 hipsolverDormqr (*C function*), 20
 hipsolverDormqr_bufferSize (*C function*), 20
 hipsolverDormtr (*C function*), 21
 hipsolverDormtr_bufferSize (*C function*), 20
 hipsolverDpotrf (*C function*), 25
 hipsolverDpotrf_bufferSize (*C function*), 24
 hipsolverDpotrfBatched (*C function*), 25
 hipsolverDpotrfBatched_bufferSize (*C function*), 25
 hipsolverDpotri (*C function*), 30
 hipsolverDpotri_bufferSize (*C function*), 29
 hipsolverDpotrs (*C function*), 31
 hipsolverDpotrs_bufferSize (*C function*), 30
 hipsolverDpotrsBatched (*C function*), 31
 hipsolverDpotrsBatched_bufferSize (*C function*), 30
 hipsolverDsyevd (*C function*), 34
 hipsolverDsyevd_bufferSize (*C function*), 34
 hipsolverDsyevj (*C function*), 38
 hipsolverDsyevj_bufferSize (*C function*), 37
 hipsolverDsyevjBatched (*C function*), 39
 hipsolverDsyevjBatched_bufferSize (*C function*), 38
 hipsolverDsygvd (*C function*), 35
 hipsolverDsygvd_bufferSize (*C function*), 35
 hipsolverDsygvj (*C function*), 40
 hipsolverDsygvj_bufferSize (*C function*), 39
 hipsolverDsytrd (*C function*), 29
 hipsolverDsytrd_bufferSize (*C function*), 28
 hipsolverDsytrf (*C function*), 26
 hipsolverDsytrf_bufferSize (*C function*), 26
 hipsolverEigMode_t (*C enum*), 15
 hipsolverEigMode_t.HIPSOLVER_EIG_MODE_NOVECTOR (*C enumerator*), 15
 hipsolverEigMode_t.HIPSOLVER_EIG_MODE_VECTOR (*C enumerator*), 15
 hipsolverEigRange_t (*C enum*), 44
 hipsolverEigRange_t.HIPSOLVER_EIG_RANGE_ALL (*C enumerator*), 44
 hipsolverEigRange_t.HIPSOLVER_EIG_RANGE_I (*C enumerator*), 44
 hipsolverEigRange_t.HIPSOLVER_EIG_RANGE_V (*C enumerator*), 44
 hipsolverEigType_t (*C enum*), 15
 hipsolverEigType_t.HIPSOLVER_EIG_TYPE_1 (*C enumerator*), 15
 hipsolverEigType_t.HIPSOLVER_EIG_TYPE_2 (*C enumerator*), 15
 hipsolverEigType_t.HIPSOLVER_EIG_TYPE_3 (*C enumerator*), 15
 hipsolverFillMode_t (*C enum*), 14
 hipsolverFillMode_t.HIPSOLVER_FILL_MODE_LOWER (*C enumerator*), 14
 hipsolverFillMode_t.HIPSOLVER_FILL_MODE_UPPER (*C enumerator*), 14
 hipsolverGesvdjInfo_t (*C type*), 44
 hipsolverGetStream (*C function*), 16
 hipsolverHandle_t (*C type*), 14
 hipsolverOperation_t (*C enum*), 14
 hipsolverOperation_t.HIPSOLVER_OP_C (*C enumerator*), 14
 hipsolverOperation_t.HIPSOLVER_OP_N (*C enumerator*), 14
 hipsolverOperation_t.HIPSOLVER_OP_T (*C enumerator*), 14
 hipsolverSetStream (*C function*), 16
 hipsolverSgebrd (*C function*), 28
 hipsolverSgebrd_bufferSize (*C function*), 28
 hipsolverSgeqrf (*C function*), 27
 hipsolverSgeqrf_bufferSize (*C function*), 27
 hipsolverSgesvd (*C function*), 36
 hipsolverSgesvd_bufferSize (*C function*), 36
 hipsolverSgesvdj (*C function*), 42
 hipsolverSgesvdj_bufferSize (*C function*), 41
 hipsolverSgesvdjBatched (*C function*), 42
 hipsolverSgesvdjBatched_bufferSize (*C function*), 41
 hipsolverSgetrf (*C function*), 26
 hipsolverSgetrf_bufferSize (*C function*), 26
 hipsolverSgetrs (*C function*), 32
 hipsolverSgetrs_bufferSize (*C function*), 31

hipsolverSideMode_t (C enum), 15
 hipsolverSideMode_t.HIPSOLVER_SIDE_LEFT
 (C enumerator), 15
 hipsolverSideMode_t.HIPSOLVER_SIDE_RIGHT
 (C enumerator), 15
 hipsolverSorgbr (C function), 19
 hipsolverSorgbr_bufferSize (C function), 19
 hipsolverSorgqr (C function), 19
 hipsolverSorgqr_bufferSize (C function), 19
 hipsolverSorgtr (C function), 20
 hipsolverSorgtr_bufferSize (C function), 20
 hipsolverSormqr (C function), 20
 hipsolverSormqr_bufferSize (C function), 20
 hipsolverSormtr (C function), 21
 hipsolverSormtr_bufferSize (C function), 20
 hipsolverSpotrf (C function), 25
 hipsolverSpotrf_bufferSize (C function), 24
 hipsolverSpotrfBatched (C function), 25
 hipsolverSpotrfBatched_bufferSize (C
 function), 25
 hipsolverSpotri (C function), 30
 hipsolverSpotri_bufferSize (C function), 29
 hipsolverSpotrs (C function), 31
 hipsolverSpotrs_bufferSize (C function), 30
 hipsolverSpotrsBatched (C function), 31
 hipsolverSpotrsBatched_bufferSize (C
 function), 30
 hipsolverSSgels (C function), 33
 hipsolverSSgels_bufferSize (C function), 33
 hipsolverSSgesv (C function), 32
 hipsolverSSgesv_bufferSize (C function), 32
 hipsolverSsyevd (C function), 34
 hipsolverSsyevd_bufferSize (C function), 34
 hipsolverSsyevj (C function), 38
 hipsolverSsyevj_bufferSize (C function), 37
 hipsolverSsyevjBatched (C function), 39
 hipsolverSsyevjBatched_bufferSize (C
 function), 38
 hipsolverSsygvd (C function), 35
 hipsolverSsygvd_bufferSize (C function), 35
 hipsolverSsygvj (C function), 40
 hipsolverSsygvj_bufferSize (C function), 39
 hipsolverSsytrd (C function), 29
 hipsolverSsytrd_bufferSize (C function), 28
 hipsolverSsytrf (C function), 26
 hipsolverSsytrf_bufferSize (C function), 26
 hipsolverStatus_t (C enum), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_ALLOC_FAILED
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_ARCH_MISMATCH
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_EXECUTION_FAILED
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_HANDLE_IS_NULLP
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_INTERNAL_ERROR
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_INVALID_ENUM
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_INVALID_VALUE
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_MAPPING_ERROR
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_NOT_INITIALIZED
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_NOT_SUPPORTED
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_SUCCESS
 (C enumerator), 14
 hipsolverStatus_t.HIPSOLVER_STATUS_UNKNOWN
 (C enumerator), 14
 hipsolverSyevjInfo_t (C type), 14
 hipsolverXgesvdjGetResidual (C function), 17
 hipsolverXgesvdjGetSweeps (C function), 17
 hipsolverXgesvdjSetMaxSweeps (C function),
 17
 hipsolverXgesvdjSetSortEig (C function), 17
 hipsolverXgesvdjSetTolerance (C function),
 17
 hipsolverXsyevjGetResidual (C function), 18
 hipsolverXsyevjGetSweeps (C function), 18
 hipsolverXsyevjSetMaxSweeps (C function), 18
 hipsolverXsyevjSetSortEig (C function), 18
 hipsolverXsyevjSetTolerance (C function), 18
 hipsolverZgebrd (C function), 28
 hipsolverZgebrd_bufferSize (C function), 28
 hipsolverZgeqrf (C function), 27
 hipsolverZgeqrf_bufferSize (C function), 27
 hipsolverZgesvd (C function), 36
 hipsolverZgesvd_bufferSize (C function), 36
 hipsolverZgesvdj (C function), 42
 hipsolverZgesvdj_bufferSize (C function), 41
 hipsolverZgesvdjBatched (C function), 42
 hipsolverZgesvdjBatched_bufferSize (C
 function), 41
 hipsolverZgetrf (C function), 26
 hipsolverZgetrf_bufferSize (C function), 26
 hipsolverZgetrs (C function), 32
 hipsolverZgetrs_bufferSize (C function), 31
 hipsolverZheevd (C function), 35
 hipsolverZheevd_bufferSize (C function), 34
 hipsolverZheevj (C function), 39
 hipsolverZheevj_bufferSize (C function), 38
 hipsolverZheevjBatched (C function), 39
 hipsolverZheevjBatched_bufferSize (C
 function), 38
 hipsolverZhegvd (C function), 36

hipsolverZhegvd_bufferSize (C function), 35
hipsolverZhegvj (C function), 40
hipsolverZhegvj_bufferSize (C function), 40
hipsolverZhetrd (C function), 29
hipsolverZhetrd_bufferSize (C function), 28
hipsolverZpotrf (C function), 25
hipsolverZpotrf_bufferSize (C function), 24
hipsolverZpotrfBatched (C function), 25
hipsolverZpotrfBatched_bufferSize (C function), 25
hipsolverZpotri (C function), 30
hipsolverZpotri_bufferSize (C function), 29
hipsolverZpotrs (C function), 31
hipsolverZpotrs_bufferSize (C function), 30
hipsolverZpotrsBatched (C function), 31
hipsolverZpotrsBatched_bufferSize (C function), 30
hipsolverZsytrf (C function), 26
hipsolverZsytrf_bufferSize (C function), 26
hipsolverZungbr (C function), 22
hipsolverZungbr_bufferSize (C function), 21
hipsolverZungqr (C function), 22
hipsolverZungqr_bufferSize (C function), 22
hipsolverZungtr (C function), 22
hipsolverZungtr_bufferSize (C function), 22
hipsolverZumqr (C function), 23
hipsolverZumqr_bufferSize (C function), 23
hipsolverZumtr (C function), 23
hipsolverZumtr_bufferSize (C function), 23
hipsolverZZgels (C function), 33
hipsolverZZgels_bufferSize (C function), 33
hipsolverZZgesv (C function), 32
hipsolverZZgesv_bufferSize (C function), 32