Package 'rlips'

January 13, 2014

Version 0.5-3

Date 2013-03-21

Title R Linear Inverse Problem Solver

Author@R c(person(``Mikko",``Orispää",email=``mikko.orispaa@oulu.fi"))

Author Mikko Orispää <mikko.orispaa@oulu.fi>

Maintainer Mikko Orispää <mikko.orispaa@oulu.fi>

Depends R (>= 2.10.0)

Description Solver for large (overdetermined) linear statistical inverse problems. Uses GPU's and OpenCL to increase performance.

License FreeBSD

Encoding UTF-8

R topics documented:

RLIPS *RLIPS*

Description

R Linear Inverse Problem Solver

Details

RLIPS is a linear solver for large (overdetermined) linear systems. It utilizes GPU's to speed up calculations. The critical routines of RLIPS are written in C and OpenCL.

RLIPS transforms the original linear problem into a upper triangular one using Givens rotations. Upper triangular system is then solved using backsubstitution.

RLIPS functions:

`rlips.init` RLIPS initialization

`rlips.dispose` RLIPS object deletion

`rlips.add` Data feeding

`rlips.solve` Problem solving

`rlips.rotate` Force Givens rotations on added data

`rlips.test`

Note

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

`rlips.add` *Add data to RLIPS problem*

Description

Add data (theory matrix, measurements and errors) into the RLIPS system row(s)-by-row(s).

Usage

```
rlips.add(e,A.data,M.data,E.data=1)
```

Arguments

`e`	Initialized RLIPS environment
`A.data`	Matrix or vector containing the theory matrix rows.
`M.data`	Matrix or vector containing the measurement matrix rows.
`E.data`	Vector or scalar containing the measurement errors (variances), or the full error covariance matrix. See details.

Details

Theory matrix rows (`A.data`) can be given as a vector (row-wise) or as a matrix.

If given as a vector, the size of the vector must be `num.rows*ncols`, where `ncols` is the number of unknowns in the problem (i.e. the number of columns in the theory matrix) and `num.rows` is the number of rows in the theory matrix `A.data`.

If given as a matrix, its dimensions must be `c(num.rows,ncols)`.

In the same way, the measurements can be given as a vector or a matrix. If given as a vector, its size nust be `num.rows*nrhs`, where `nrhs` is the number of columns in the measurement vector. If it is given as a matrix its size must be `c(num.rows,nrhs)`.

The error variances can be given as a vector (size `num.rows`) or if the error variance is constant for all measurements added into RLIPS, it can also be given as a single scalar value. If error is omitted scalar value 1.0 will be used.

If the full error covariance matrix is given, it must be a symmetric positive-definite square matrix with size `num.rows`-by-`num.rows`.

Value

None.

Author(s)

Mikko Orispaa (University of Oulu, Finland) `<mikko.orispaa@oulu.fi>`

`rlips.copy` *Copy RLIPS environment*

Description

Copy RLIPS environment into a new environment

Usage

```
rlips.copy(oenv)
```

Arguments

`oenv`	RLIPS environment

Value

RLIPS environment which is a copy of `oenv`.

Author(s)

Mikko Orispaa (University of Oulu, Finland) `<mikko.orispaa@oulu.fi>`

rlips.dispose *Delete rlips environment*

Description

Deletes the rlips enviroment's internal variables and the files used by the environment.

Usage

```
rlips.dispose(e)
```

Arguments

e	rlips environment to be deleted

Details

`rlips.dispose` will delete all variables in rlips environment `h`.

Value

None. `e` will be an empty environment afterwards.

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

rlips.get.data *Fetch the rotated data from the GPU*

Description

Fetches the target matrix R and the target vector Y from the GPU.

Usage

```
rlips.get.data(e)
```

Arguments

e	Initialized RLIPS environment

Details

The rotated system matrices are fetched from the GPU. The upper triangular target matrix R is put into `e$R.mat` and the corresponding target vector Y is put into `e$Y.mat`.

Note that this routine is called every time the command `rlips.solve` is used, so there is no reason to use this in normal operation.

Value

None.

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

`rlips.init` *Initialize rlips environment*

Description

Initializes a new rlips environment.

Usage

```
rlips.init(ncols,nrhs,type=s,nbuf=ncols,workgroup.size=128)
```

Arguments

`ncols`	Integer giving the number of unknowns, i.e. the number of columns in the theory matrix.
`nrhs`	Integer giving the number of columns in the measurement matrix.
`type`	Problem numerical type. One of the following character strings: **'s'** Single precision real **'c'** Single precision complex
`nbuf`	Size (number of rows) of the rotation buffer. Default is the number of unknowns, i.e. default rotation buffer is of the same size as the problem after rotations. Size of the buffer can have a significant impact on performance. The optimal size of the buffer depends on the problem and also on computer hardware. Small buffers conserve computer memory but tend to slow rlips down.
`workgroup.size`	Size of the OpenCL workgroup. Optimal size depends on GPU hardware. Usually, This should be of form `2**n`, where `n = 2,3,...,10`. Typical optimal value for NVIDIA GPU's is 128 and for AMD GPU's 256. Your mileage may vary.

Value

R environment. This environment contains a number of internal variables needed by `rrlips`. After the rlips problem is solved (see `rlips.solve`) it also contains the solution and (if calculated) posteriori covariance matrix.

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

See Also

`rlips.dispose`, `rlips.add`, `rlips.solve`.

rlips.rotate *Force Givens rotation*

Description

Rotates all unrotated data of a rlips environment

Usage

```
rlips.rotate(e)
```

Arguments

e	Existing rlips environment with unrotated data

Details

`rlips.rotate` forces the Givens rotations on unrotated rlips data. Especially for large problems this can reduce the memory consumption and solution time remarkably. On the other hand, overusing this command will decrease the performance.

The size of the rotation buffer can be controlled by `buffersize`. The default `ncols` should be good for most situations.

Value

None. The unrotated data is rotated into rlips system.

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

rlips.solve *Solve rlips problem*

Description

Solve the problem fed into rlips environment.

Usage

```
rlips.solve(e,calculate.covariance=FALSE,full.covariance=FALSE)
```

Arguments

e	An existing rlips environment with enough data feeded in.
calculate.covariance	Flag for calculating the posteriori covariance matrix.
full.covariance	Flag for calculating the full posteriori covariance matrix..

Details

The calculation of the posteriori covariance matrix is controlled by the two flags `calculate.covariance` and `full.covariance`.

If `calculate.covariance=FALSE` the posteriori covariance matrix is not calculated at all.

If `calculate.covariance=TRUE`, then if `full.covariance=TRUE` the full posteriori covariance matrix is calculated. If `full.covariance=FALSE`, only the posteriori variances (i.e. the diagonal of the posteriori covariance matrix) is calculated. This is much faster than calculating the full matrix!

Value

None. The solution is written into rlips environment variable `e$solution`. The posteriori covariance matrix (if calculated) is written into environment variable `e$covariance`.

Note

Rlips does not check that there exists enough data in the problem, nor that the problem is solvable (i.e. there exists enough linearly independent data rows in the theory matrix).

This routine calls first rlips.get.data, which fetches the rotated system from the GPU. The rotated upper triangular target matrix R is put into `e$R.mat` and the corresponding rotated target vector into `e$Y.mat`.

Author(s)

Mikko Orispaa (University of Oulu, Finland) `<mikko.orispaa@oulu.fi>`

`rlips.test` *RLIPS test*

Description

Simple test to check the speed and accuracy of RLIPS.

Usage

```
rlips.test(type,size,buffersize=size[2],loop=1,wg.size=128,return.data=FALSE,averaging.fun=mean)
```

Arguments

`type`	Type of the problem. `s` for single precision real, or `c` for single precision complex.
`size`	2-vector holding the size (rows,columns) of the test problem.
`buffersize`	RLIPS rotation buffersize. The default is the number of columns/unknowns of the problem.
`loop`	Number of tests performed. Default is 1. If `loop` is larger than 1, the results are averaged using the function `averaging.fun`.
`wg.size`	OpenCL workgroup size.
`return.data`	If `TRUE`, the rotated system matrices are returned.
`averaging.fun`	Function used in averaging the results, if `loop` is greater than 1. Default is arithmetic mean.

Details

This test routine can (and should) be used to trim the rotation buffer and OpenCL workgroup sizes for the used hardware.

Value

Returns a list with components:

time Elapsed time for solving the problem using RLIPS.

accuracy Maximum absolute error.

Gflops Approximate Gigaflops (floating point operations per second).

R If `return.data=TRUE`, the rotated upper triangular target matrix.

Y If `return.data=TRUE`, the rotated target vector.

Author(s)

Mikko Orispaa (University of Oulu, Finland) <mikko.orispaa@oulu.fi>

Index