Multi-GPU symmetric Toeplitz Eigenvalue Extractor

Generated by Doxygen 1.5.6

Wed Dec 10 08:05:20 2008
# Contents

1 Class Index .................................................................................................................. 1
   1.1 Class List ................................................................................................................ 1

2 File Index ....................................................................................................................... 3
   2.1 File List ................................................................................................................... 3

3 Class Documentation ..................................................................................................... 5
   3.1 exec_window_entry_struct Struct Reference ......................................................... 5
      3.1.1 Detailed Description ....................................................................................... 5
      3.1.2 Member Data Documentation ....................................................................... 5
         3.1.2.1 interval_index ....................................................................................... 5
         3.1.2.2 copy_stream ......................................................................................... 5
   3.2 extract_device_info_struct Struct Reference ...................................................... 6
      3.2.1 Detailed Description ....................................................................................... 7
      3.2.2 Member Data Documentation ....................................................................... 7
         3.2.2.1 window_size ......................................................................................... 7
         3.2.2.2 window_used ....................................................................................... 7
         3.2.2.3 exec_window ......................................................................................... 7
         3.2.2.4 exec_window_gpu ............................................................................... 7
         3.2.2.5 exec_stream ......................................................................................... 7
         3.2.2.6 device_prop ......................................................................................... 7
         3.2.2.7 gpu_t ....................................................................................................... 7
         3.2.2.8 gpu_p ....................................................................................................... 8
         3.2.2.9 gpu_q ....................................................................................................... 8
         3.2.2.10 gpu_workspace .................................................................................. 8
         3.2.2.11 gpu_m .................................................................................................... 8
         3.2.2.12 cpu_p .................................................................................................... 8
         3.2.2.13 cpu_q .................................................................................................... 8
         3.2.2.14 cpu_alpha ........................................................................................... 8
4.6 toeplitz.h File Reference ........................................... 34
4.6.1 Detailed Description ............................................. 35
4.6.2 Define Documentation ........................................... 35
   4.6.2.1 _axpy ............................................... 35
   4.6.2.2 _dot ............................................... 35
   4.6.2.3 _gemm ............................................... 35
   4.6.2.4 _gemv ............................................... 36
   4.6.2.5 _nrm2 ............................................... 36
   4.6.2.6 _scal ............................................... 36
   4.6.2.7 _spmv ............................................... 36
   4.6.2.8 _spsv ............................................... 36
   4.6.2.9 _stev ............................................... 36
   4.6.2.10 align .............................................. 36
   4.6.2.11 blockSize ............................................ 36
   4.6.2.12 FALSE .............................................. 36
   4.6.2.13 half_warp ........................................... 36
   4.6.2.14 half_warp_mask ..................................... 36
   4.6.2.15 INTERVAL_NONE .................................... 37
   4.6.2.16 print_vector_gpu .................................... 37
   4.6.2.17 SINGLE ............................................. 37
   4.6.2.18 T ................................................... 37
   4.6.2.19 TRUE ............................................... 37
   4.6.2.20 Tsf ................................................ 37
4.6.3 Typedef Documentation ......................................... 37
   4.6.3.1 exec_window_entry .................................... 37
   4.6.3.2 extract_device_info .................................. 37
   4.6.3.3 extract_global_info .................................. 37
   4.6.3.4 interval_cpu_info .................................... 37
4.6.4 Function Documentation ........................................ 37
   4.6.4.1 __align__ ............................................. 37
   4.6.4.2 __align__ ............................................. 38
   4.6.4.3 calc_eigen_intervals ................................ 38
   4.6.4.4 extract_eigenvalues ................................ 38
   4.6.4.5 levinson ............................................ 39
   4.6.4.6 print_matrix ........................................ 39
   4.6.4.7 print_vector ........................................ 40
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6.4.8</td>
<td>readInput</td>
<td>40</td>
</tr>
<tr>
<td>4.6.4.9</td>
<td>sort_eigenvalues</td>
<td>40</td>
</tr>
<tr>
<td>4.6.4.10</td>
<td>toeplitz_gershgorin</td>
<td>40</td>
</tr>
<tr>
<td>4.6.4.11</td>
<td>toeplitz_mult</td>
<td>41</td>
</tr>
<tr>
<td>4.6.5</td>
<td>Variable Documentation</td>
<td>41</td>
</tr>
<tr>
<td>4.6.5.1</td>
<td>interval_gpu_info</td>
<td>41</td>
</tr>
<tr>
<td>4.6.5.2</td>
<td>tridiagonal_entry</td>
<td>41</td>
</tr>
<tr>
<td>4.7</td>
<td>toeplitz_kernel.cu File Reference</td>
<td>42</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Detailed Description</td>
<td>42</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Function Documentation</td>
<td>42</td>
</tr>
<tr>
<td>4.7.2.1</td>
<td>levinson</td>
<td>42</td>
</tr>
<tr>
<td>4.7.2.2</td>
<td>memcpy4</td>
<td>43</td>
</tr>
<tr>
<td>4.7.2.3</td>
<td>reorthogonalize</td>
<td>43</td>
</tr>
<tr>
<td>4.7.2.4</td>
<td>si2w</td>
<td>43</td>
</tr>
<tr>
<td>4.7.2.5</td>
<td>si2w_parallel</td>
<td>44</td>
</tr>
<tr>
<td>4.8</td>
<td>utils.c File Reference</td>
<td>45</td>
</tr>
<tr>
<td>4.8.1</td>
<td>Detailed Description</td>
<td>45</td>
</tr>
<tr>
<td>4.8.2</td>
<td>Function Documentation</td>
<td>45</td>
</tr>
<tr>
<td>4.8.2.1</td>
<td>print_matrix</td>
<td>45</td>
</tr>
<tr>
<td>4.8.2.2</td>
<td>print_vector</td>
<td>45</td>
</tr>
<tr>
<td>4.8.2.3</td>
<td>readInput</td>
<td>46</td>
</tr>
</tbody>
</table>
Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

- exec_window_entry_struct ........................................... 5
- extract_device_info_struct ........................................ 6
- extract_global_info_struct ......................................... 10
- interval_cpu_info_struct .......................................... 14
- interval_node_struct .............................................. 18
- sort_aux_struct .................................................... 19
Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

- cpu_eigen.c (Symmetric Toeplitz matrix eigenvalue extractor (sequential CPU version) ) 21
- eigen.c (Eigenvalue related functions: interval extraction, convergence checking and sorting ) 23
- gpu_eigen.cu (Symmetric Toeplitz matrix eigenvalue extractor (multi-GPU version) ) 26
- gpu_vector_kernel.cu (Kernel functions for vector operations ) 29
- toeplitz.c (Toeplitz matrix related functions ) 32
- toeplitz.h (Main project header file ) 34
- toeplitz_kernel.cu (Kernel functions for Levinson algorithm and shift and invert 2-way Lanczos ) 42
- utils.c (Utility functions for eigenvalue extraction tool ) 45
Chapter 3

Class Documentation

3.1 exec_window_entry_struct Struct Reference

#include <toeplitz.h>

Public Attributes

• int interval_index
  
  Associated interval index or INTERVAL_NONE.

• cudaStream_t copy_stream
  
  Asynchronous copy stream object.

3.1.1 Detailed Description

Definition at line 151 of file toeplitz.h.

3.1.2 Member Data Documentation

3.1.2.1 int exec_window_entry_struct::interval_index

Associated interval index or INTERVAL_NONE.
Definition at line 152 of file toeplitz.h.

3.1.2.2 cudaStream_t exec_window_entry_struct::copy_stream

Asynchronous copy stream object.
Definition at line 153 of file toeplitz.h.

The documentation for this struct was generated from the following file:

• toeplitz.h
Public Attributes

• unsigned int window_size
  Number of intervals executed simultaneously.

• unsigned int window_used
  Number of active intervals in the execution window.

• exec_window_entry * exec_window
  Indices of intervals in execution. Set to INTERVAL_NONE if no interval associated.

• interval_gpu_info * exec_window_gpu
  Pointers to interval descriptors stored in global GPU memory.

• cudaStream_t exec_stream
  Asynchronous copy stream object for execution window data.

• struct cudaDeviceProp * device_prop
  CUDA device properties.

• T * gpu_t
  GPU array storing problem data (main diagonal not included, passed as an argument).

• T * gpu_p
• T * gpu_q
  GPU arrays for Krylov subspace matrices (for all execution window intervals).

• T * workspace
  GPU arrays for auxiliar workspace (for all execution window intervals).

• tridiagonal_entry * gpu_m
  GPU arrays for storing tridiagonal results (for all execution window intervals).

• T * cpu_p
• T * cpu_q
  CPU arrays for Krylov subspace matrices (for all execution window intervals).

• T * cpu_alpha
• T * cpu_beta
  CPU arrays for symmetric subspace tridiagonal results (for all execution window intervals).

• T * cpu_gamma
• T * cpu_delta
  CPU arrays for skew-symmetric subspace tridiagonal results (for all execution window intervals).
3.2 extract_device_info_struct Struct Reference

- tridiagonal_entry * cpu_m
  CPU arrays for skew-symmetric subspace tridiagonal results (for all execution window intervals).

3.2.1 Detailed Description

Definition at line 157 of file toeplitz.h.

3.2.2 Member Data Documentation

3.2.2.1 unsigned int extract_device_info_struct::window_size

Number of intervals executed simultaneously.
Definition at line 160 of file toeplitz.h.

3.2.2.2 unsigned int extract_device_info_struct::window_used

Number of active intervals in the execution window.
Definition at line 161 of file toeplitz.h.

3.2.2.3 exec_window_entry* extract_device_info_struct::exec_window

Indices of intervals in execution. Set to INTERVAL_NONE if no interval associated.
Definition at line 162 of file toeplitz.h.

3.2.2.4 interval_gpu_info* extract_device_info_struct::exec_window_gpu

Pointers to interval descriptors stored in global GPU memory.
Definition at line 163 of file toeplitz.h.

3.2.2.5 cudaStream_t extract_device_info_struct::exec_stream

Asynchronous copy stream object for execution window data.
Definition at line 164 of file toeplitz.h.

3.2.2.6 struct cudaDeviceProp* extract_device_info_struct::device_prop [read]

CUDA device properties.
Definition at line 167 of file toeplitz.h.

3.2.2.7 T* extract_device_info_struct::gpu_t

GPU array storing problem data (main diagonal not included, passed as an argument).
Definition at line 170 of file toeplitz.h.
3.2.2.8 \textit{T* extract_device_info struct::gpu\_p}

Definition at line 171 of file toeplitz.h.

3.2.2.9 \textit{T * extract_device_info_struct::gpu\_q}

GPU arrays for Krylov subspace matrices (for all execution window intervals).
Definition at line 171 of file toeplitz.h.

3.2.2.10 \textit{T* extract_device_info_struct::gpu\_workspace}

GPU arrays for auxiliar workspace (for all execution window intervals).
Definition at line 172 of file toeplitz.h.

3.2.2.11 \textit{tridiagonal_entry* extract_device_info_struct::gpu\_m}

GPU arrays for storing tridiagonal results (for all execution window intervals).
Definition at line 173 of file toeplitz.h.

3.2.2.12 \textit{T* extract_device_info_struct::cpu\_p}

Definition at line 175 of file toeplitz.h.

3.2.2.13 \textit{T * extract_device_info_struct::cpu\_q}

CPU arrays for Krylov subspace matrices (for all execution window intervals).
Definition at line 175 of file toeplitz.h.

3.2.2.14 \textit{T* extract_device_info_struct::cpu\_alpha}

Definition at line 176 of file toeplitz.h.

3.2.2.15 \textit{T * extract_device_info_struct::cpu\_beta}

CPU arrays for symmetric subspace tridiagonal results (for all execution window intervals).
Definition at line 176 of file toeplitz.h.

3.2.2.16 \textit{T* extract_device_info_struct::cpu\_gamma}

Definition at line 177 of file toeplitz.h.

3.2.2.17 \textit{T * extract_device_info_struct::cpu\_delta}

CPU arrays for skew-symmetric subspace tridiagonal results (for all execution window intervals).
Definition at line 177 of file toeplitz.h.
3.2.18 tridiagonal_entry extract_device_info_struct::cpu_m

CPU arrays for skew-symmetric subspace tridiagonal results (for all execution window intervals).
Definition at line 178 of file toeplitz.h.
The documentation for this struct was generated from the following file:

  • toeplitz.h
3.3  
\texttt{extract\_global\_info\_struct} Struct Reference

\texttt{#include <toeplitz.h>}

\textbf{Public Attributes}

- \texttt{unsigned int num\_intervals}
  
  \textit{Number of total intervals.}

- \texttt{unsigned int max\_eig\_interval}
  
  \textit{Maximum number of eigenvalues per interval.}

- \texttt{unsigned int max\_k}
  
  \textit{Maximum Krylov subspace size in all intervals.}

- \texttt{unsigned int max\_retry\_count}
  
  \textit{Maximum of retries of each interval.}

- \texttt{unsigned int inc\_k}
  
  \textit{Subspace size increment in each iteration.}

- \texttt{unsigned int n}
  
  \textit{Size of the problem.}

- \texttt{T * t}
  
  \textit{Problem data values defining a symmetric Toeplitz matrix.}

- \texttt{T tolerance}
  
  \textit{Required tolerance for eigenvalue extraction.}

- \texttt{interval\_cpu\_info * cpu}
  
  \textit{CPU-side interval information.}

- \texttt{interval\_gpu\_info * gpu}
  
  \textit{GPU-side interval information (stored in CPU memory).}

- \texttt{unsigned int next\_interval}
  
  \textit{Next interval to be processed.}

- \texttt{unsigned int global\_window\_used}
  
  \textit{Number of active intervals in execution windows of all devices.}

- \texttt{unsigned int num\_devices}
  
  \textit{Number of CUDA-enabled devices available.}

- \texttt{extract\_device\_info * device}
  
  \textit{Per-device eigenvalue extraction information.}

- \texttt{unsigned int size\_pq}
Size, in $T$ elements, of Krylov subspace matrices for each interval.

- **unsigned int** `size_workspace`
  Size, in $T$ elements, of each interval’s workspace.

- **unsigned int** `size_toeplitz_data`
  Size, in bytes, of problem data defining a symmetric Toeplitz matrix.

- **unsigned int** `size_smem`
  Required preallocated shared memory by GPU kernels.

### 3.3.1 Detailed Description

Definition at line 183 of file toeplitz.h.

### 3.3.2 Member Data Documentation

#### 3.3.2.1 `unsigned int extract_global_info_struct::num_intervals`

Number of total intervals.
Definition at line 186 of file toeplitz.h.

#### 3.3.2.2 `unsigned int extract_global_info_struct::max_eig_interval`

Maximum number of eigenvalues per interval.
Definition at line 187 of file toeplitz.h.

#### 3.3.2.3 `unsigned int extract_global_info_struct::max_k`

Maximum Krylov subspace size in all intervals.
Definition at line 188 of file toeplitz.h.

#### 3.3.2.4 `unsigned int extract_global_info_struct::max_retry_count`

Maximum of retries of each interval.
Definition at line 189 of file toeplitz.h.

#### 3.3.2.5 `unsigned int extract_global_info_struct::inc_k`

Subspace size increment in each iteration.
Definition at line 190 of file toeplitz.h.
3.3.2.6 unsigned int extract_global_info_struct::n
Size of the problem.
Definition at line 193 of file toeplitz.h.

3.3.2.7 T* extract_global_info_struct::t
Problem data values defining a symmetric Toeplitz matrix.
Definition at line 194 of file toeplitz.h.

3.3.2.8 T extract_global_info_struct::tolerance
Required tolerance for eigenvalue extraction.
Definition at line 195 of file toeplitz.h.

3.3.2.9 interval_cpu_info* extract_global_info_struct::cpu
CPU-side interval information.
Definition at line 198 of file toeplitz.h.

3.3.2.10 interval_gpu_info* extract_global_info_struct::gpu
GPU-side interval information (stored in CPU memory).
Definition at line 199 of file toeplitz.h.

3.3.2.11 unsigned int extract_global_info_struct::next_interval
Next interval to be processed.
Definition at line 200 of file toeplitz.h.

3.3.2.12 unsigned int extract_global_info_struct::global_window_used
Number of active intervals in execution windows of all devices.
Definition at line 201 of file toeplitz.h.

3.3.2.13 unsigned int extract_global_info_struct::num_devices
Number of CUDA-enabled devices available.
Definition at line 204 of file toeplitz.h.

3.3.2.14 extract_device_info* extract_global_info_struct::device
Per-device eigenvalue extraction information.
Definition at line 205 of file toeplitz.h.
3.3.2.15 unsigned int extract_global_info_struct::size_pq

Size, in T elements, of Krylov subspace matrices for each interval.
Definition at line 208 of file toeplitz.h.

3.3.2.16 unsigned int extract_global_info_struct::size_workspace

Size, in T elements, of each interval’s workspace.
Definition at line 209 of file toeplitz.h.

3.3.2.17 unsigned int extract_global_info_struct::size_toeplitz_data

Size, in bytes, of problem data defining a symmetric Toeplitz matrix.
Definition at line 210 of file toeplitz.h.

3.3.2.18 unsigned int extract_global_info_struct::size_smem

Required preallocated shared memory by GPU kernels.
Definition at line 211 of file toeplitz.h.
The documentation for this struct was generated from the following file:

  * toeplitz.h
3.4 interval_cpu_info_struct Struct Reference

#include <toeplitz.h>

Public Attributes

- T lb
- T ub
  
  Lower and upper bounds of the interval.

- unsigned int num_eigs
  
  Number of eigenvalues in the interval.

- T * p
- T * q
  
  Symmetric and skew-symmetric Lanczos subspace bases.

- T * ls
- T * la
- T * vs
- T * va
  
  Eigenvalues and eigenvectors extracted from symmetric and skew-symmetric subspaces.

- T * alpha
- T * beta
- T * gamma
- T * delta
  
  Tridiagonal entries from symmetric and skew-symmetric subspaces.

- tridiagonal_entry * m
  
  Tridiagonal entries from symmetric and skew-symmetric subspaces. Used for asynchronous copies.

- int eigen_left
  
  Number of eigenvalues remaining to be extracted.

- unsigned int extracted_s
- unsigned int extracted_a
  
  Number of eigenvalues already extracted from symmetric and skew-symmetric subspaces.

- unsigned int converged_s
- unsigned int converged_a
  
  Number of already converged eigenvalues from symmetric and skew-symmetric subspaces.

- unsigned int retry_count
  
  Number of times that the interval has been retried.

3.4.1 Detailed Description

Definition at line 117 of file toeplitz.h.
3.4.2 Member Data Documentation

3.4.2.1 T interval_cpu_info_struct::lb

Definition at line 120 of file toeplitz.h.

3.4.2.2 T interval_cpu_info_struct::ub

Lower and upper bounds of the interval.
Definition at line 120 of file toeplitz.h.

3.4.2.3 unsigned int interval_cpu_info_struct::num_eigs

Number of eigenvalues in the interval.
Definition at line 121 of file toeplitz.h.

3.4.2.4 T* interval_cpu_info_struct::p

Definition at line 124 of file toeplitz.h.

3.4.2.5 T* interval_cpu_info_struct::q

Symmetric and skew-symmetric Lanczos subspace bases.
Definition at line 124 of file toeplitz.h.

3.4.2.6 T* interval_cpu_info_struct::ls

Definition at line 125 of file toeplitz.h.

3.4.2.7 T* interval_cpu_info_struct::la

Definition at line 125 of file toeplitz.h.

3.4.2.8 T* interval_cpu_info_struct::vs

Definition at line 125 of file toeplitz.h.

3.4.2.9 T* interval_cpu_info_struct::va

Eigenvalues and eigenvectors extracted from symmetric and skew-symmetric subspaces.
Definition at line 125 of file toeplitz.h.

3.4.2.10 T* interval_cpu_info_struct::alpha

Definition at line 126 of file toeplitz.h.
3.4.2.11  \( T \ast interval_cpu_info_struct::beta \)
Definition at line 126 of file toeplitz.h.

3.4.2.12  \( T \ast interval_cpu_info_struct::gamma \)
Definition at line 126 of file toeplitz.h.

3.4.2.13  \( T \ast interval_cpu_info_struct::delta \)
Tridiagonal entries from symmetric and skew-symmetric subspaces.
Definition at line 126 of file toeplitz.h.

3.4.2.14  tridiagonal_entry* interval_cpu_info_struct::m
Tridiagonal entries from symmetric and skew-symmetric subspaces. Used for asynchronous copies.
Definition at line 127 of file toeplitz.h.

3.4.2.15  int interval_cpu_info_struct::eigen_left
Number of eigenvalues remaining to be extracted.
Definition at line 129 of file toeplitz.h.

3.4.2.16  unsigned int interval_cpu_info_struct::extracted_s
Definition at line 130 of file toeplitz.h.

3.4.2.17  unsigned int interval_cpu_info_struct::extracted_a
Number of eigenvalues already extracted from symmetric and skew-symmetric subspaces.
Definition at line 130 of file toeplitz.h.

3.4.2.18  unsigned int interval_cpu_info_struct::converged_s
Definition at line 131 of file toeplitz.h.

3.4.2.19  unsigned int interval_cpu_info_struct::converged_a
Number of already converged eigenvalues from symmetric and skew-symmetric subspaces.
Definition at line 131 of file toeplitz.h.

3.4.2.20  unsigned int interval_cpu_info_struct::retry_count
Number of times that the interval has been retried.
Definition at line 132 of file toeplitz.h.
The documentation for this struct was generated from the following file:

- toeplitz.h
3.5 interval_node_struct Struct Reference

**Public Attributes**

- double lb
- double ub
- integer eigs
- integer prev_eigs
- integer post_eigs
- struct interval_node_struct * next

3.5.1 Detailed Description

Definition at line 32 of file eigen.c.

3.5.2 Member Data Documentation

3.5.2.1 double interval_node_struct::lb

Definition at line 33 of file eigen.c.

3.5.2.2 double interval_node_struct::ub

Definition at line 33 of file eigen.c.

3.5.2.3 integer interval_node_struct::eigs

Definition at line 34 of file eigen.c.

3.5.2.4 integer interval_node_struct::prev_eigs

Definition at line 34 of file eigen.c.

3.5.2.5 integer interval_node_struct::post_eigs

Definition at line 34 of file eigen.c.

3.5.2.6 struct interval_node_struct * interval_node_struct::next

[read]

Definition at line 35 of file eigen.c.

The documentation for this struct was generated from the following file:

- eigen.c
3.6 sort_aux_struct_ Struct Reference

Public Attributes

- T el
- unsigned int index

3.6.1 Detailed Description

Definition at line 42 of file eigen.c.

3.6.2 Member Data Documentation

3.6.2.1 T sort_aux_struct_::el

Definition at line 43 of file eigen.c.

3.6.2.2 unsigned int sort_aux_struct_::index

Definition at line 44 of file eigen.c.

The documentation for this struct was generated from the following file:

- eigen.c
Chapter 4

File Documentation

4.1 cpu_eigen.c File Reference

Symmetric Toeplitz matrix eigenvalue extractor (sequential CPU version).

#include <math.h>
#include <time.h>
#include <stdio.h>
#include "toeplitz.h"
#include "cmdline_cpu.h"

Functions

• void parse_cmdline (int argc, char *argv[], struct gengetopt_args_info_cpu *args_info)
• int si2w_cpu (unsigned int n, const T *t, T ia, T ib, T sigma, unsigned int nEig, unsigned int max_k, T tolerance, T *l_out, T *v_out)
• int main (int argc, char *argv[])

Variables

• static struct gengetopt_args_info_cpu args_info

4.1.1 Detailed Description

Symmetric Toeplitz matrix eigenvalue extractor (sequential CPU version).

Author:

Leandro Graciá Gil, leagragi@inf.upv.es

Date:

18/11/08

Definition in file cpu_eigen.c.
4.1.2 Function Documentation

4.1.2.1 int main (int argc, char * argv[])

Definition at line 214 of file cpu_eigen.c.

4.1.2.2 void parse_cmdline (int argc, char * argv[], struct gengetopt_args_info_cpu * args_info)

Parse command line commands into args_info structure (uses gengetopt).

Parameters:
- argc Number of params in command line.
- argv Params of command line.
- args_info Structure to be filled with parsed input params.

Definition at line 49 of file cpu_eigen.c.

4.1.2.3 int si2w_cpu (unsigned int n, const T * t, T ia, T ib, T sigma, unsigned int nEig, unsigned int max_k, T tolerance, T * l_out, T * v_out)

CPU version of the Shift-and-invert 2-way Lanczos algorithm.

Note:
Extracted eigenvalues are not sorted.

Parameters:
- n Size of t vector.
- t Symmetric Toeplitz matrix data in vector form.
- ia Start of the eigenvalue extraction interval.
- ib End of the eigenvalue extraction interval.
- sigma Shift value used for eigenvalue extraction.
- nEig Number of required eigenvalues in the interval.
- max_k Maximum size of Krylov subspace.
- tolerance Tolerance threshold used to check eigenvalue convergence.
- l_out Vector of size nEig containing extracted eigenvalues.
- v_out Matrix of size n nEig (column order) containing extracted eigenvectors.

Returns:
Number of eigenvalues extracted.

Definition at line 84 of file cpu_eigen.c.

4.1.3 Variable Documentation

4.1.3.1 struct gengetopt_args_info_cpu args_info [static]

Definition at line 40 of file cpu_eigen.c.
4.2 eigen.c File Reference

Eigenvalue related functions: interval extraction, convergence checking and sorting.
#include "toeplitz.h"

Classes

• struct interval_node_struct
• struct sort_aux_struct_

Typedefs

• typedef struct interval_node_struct interval_node
• typedef struct sort_aux_struct_ sort_aux_struct

Functions

• void calc_intervals (interval_node *ilist, integer *n, double *t, double *lbg, double *ubg, integer *maxEig_interval)
• static int sort_eigenvalues_aux (const void *p1, const void *p2)
• void sort_eigenvalues (unsigned int extracted, unsigned int n, T *el, T *ev)
• interval_cpu_info * calc_eigen_intervals (T *t, unsigned int n, T lbg, T ubg, unsigned int max_eig_interval, unsigned int *num_intervals)
• unsigned int extract_eigenvalues (unsigned int n, unsigned int k, unsigned int max_k, unsigned int p_lda, unsigned int *converged, unsigned int *extracted, unsigned int required, T a, T b, const T *t, const T *p, const T *alpha, const T *beta, T tolerance, T *l, T *v)

4.2.1 Detailed Description

Eigenvalue related functions: interval extraction, convergence checking and sorting.

Author:
Leandro Graciá Gil, leagragi@inf.upv.es

Date:
18/11/08

Definition in file eigen.c.
4.2.2 Type défini Documentation

4.2.2.1 typedef struct interval_node_struct interval_node

4.2.2.2 typedef struct sort_aux_struct_ sort_aux_struct

4.2.3 Fonction Documentation

4.2.3.1 interval_cpu_info* calc_eigen_intervals (T * t, unsigned int n, T lbg, T ubg, unsigned int max_eig_interval, unsigned int * num_intervals)

Calculates intervals containing approximately the same number of eigenvalues along a given interval. Only interval-specific information is written in the array. All other data is initialized to 0.

Parameters:

- $t$ Vector defining a symmetric Toeplitz matrix whose eigenvalue intervals will be calculated.
- $n$ Size of vector $t$.
- $lbg$ Lower bound of the initial interval to be subdivided (usually Gershgorin disc).
- $ubg$ Upper bound of the initial interval to be subdivided (usually Gershgorin disc).
- $max_eig_interval$ Maximum number of eigenvalues per interval.
- $num_intervals$ Number of extracted intervals.

Returns:

- Pointer to an array of size $num_intervals$ containing interval info. NULL in case of error.

Definition at line 105 of file eigen.c.

4.2.3.2 void calc_intervals (interval_node * ilist, integer * n, double * t, double * lbg, double * ubg, integer * maxEig_interval)

4.2.3.3 unsigned int extract_eigenvalues (unsigned int n, unsigned int k, unsigned int max_k, unsigned int p_lda, unsigned int * converged, unsigned int * extracted, unsigned int required, T a, T b, const T * t, const T * p, const T * alpha, const T * beta, T tolerance, T * l, T * y)

Check convergence and extract eigenvalues of a symmetric Toeplitz matrix given a tridiagonal shift-and-invert Lanczos approximation.

Note:

- sigma shift value is assumed to be the halfway between $a$ and $b$.

Parameters:

- $n$ Size of the Toeplitz matrix defined by $t$ and number of rows of $p$.
- $k$ Current size of Krylov subspace and number of columns of $p$.
- $max_k$ Maximum allowed size of the Krylov subspace.
- $p_lda$ Leading dimension of Krylov subspace vectors stored in $p$.
- $converged$ Pointer to number of converged eigenvalues. Just initialize to 0 on start and let the function use it.
4.2 eigen.c File Reference

- **extracted** Pointer to number of currently extracted eigenvalues.
- **required** Number of eigenvalues that should be extracted.
- **a** Start of the interval whose eigenvalues are being extracted.
- **b** End of the interval whose eigenvalues are being extracted.
- **t** Vector of size \( n \) defining the symmetric Toeplitz matrix.
- **p** Matrix of size \( n \times k \) containing Lanczos vectors.
- **alpha** Vector of size \( k \) containing the main diagonal of the symmetric tridiagonal matrix.
- **beta** Vector of size \( k-1 \) containing the second diagonal of the symmetric tridiagonal matrix.
- **tolerance** Tolerance threshold used to check eigenvalue convergence.
- **l** Output vector where up to **required** eigenvalues will be stored.
- **v** Output matrix where up to **required** eigenvectors will be stored (column order).

**Returns:**

Number of eigenvalues extracted during this call.

Definition at line 174 of file eigen.c.

4.2.3.4 **void sort_eigenvalues (unsigned int extracted, unsigned int n, T *el, T *ev)**

Sort eigenvalues and eigenvectors using quicksort algorithm.

**Parameters:**

- **extracted** Number of extracted eigenvalues.
- **n** Size of eigenvectors.
- **el** Pointer to vector containing eigenvalues.
- **ev** Pointer to matrix containing eigenvectors.

Definition at line 68 of file eigen.c.

4.2.3.5 **static int sort_eigenvalues_aux (const void *p1, const void *p2) [static]**

Auxiliar quicksort comparison method for the **sort_eigenvalues** function. Check stdlib qsort documentation for details.

Definition at line 51 of file eigen.c.
# 4.3 gpu_eigen.cu File Reference

Symmetric Toeplitz matrix eigenvalue extractor (multi-GPU version).

```c
#include <stdio.h>
#include <assert.h>
#include <time.h>
#include "toeplitz.h"
#include "cmdline_gpu.h"
#include "toeplitz_kernel.cu"
```

## Functions

- `void parse_cmdline (int argc, char *argv[], struct gengetopt_args_info_gpu *args_info)`
- `void assign_interval (extract_global_info *global, unsigned int num_device, unsigned int window_pos, unsigned int num_interval, unsigned int start_k)`
- `void delete_extract_info (extract_global_info *global)`
- `int calc_memory_parameters (extract_global_info *global, unsigned int max_window_size)`
- `void prepare_devices (extract_global_info *global)`
- `int main (int argc, char *argv[])`

## Variables

- `static struct gengetopt_args_info_gpu args_info`

# 4.3.1 Detailed Description

Symmetric Toeplitz matrix eigenvalue extractor (multi-GPU version).

**Author:**

Leandro Graciá Gil, leagragi@inf.upv.es

**Date:**

18/11/08

Definition in file `gpu_eigen.cu`.

# 4.3.2 Function Documentation

## 4.3.2.1 void assign_interval (extract_global_info *global, unsigned int num_device, unsigned int window_pos, unsigned int num_interval, unsigned int start_k)

Assign a given interval to the current execution window. Detaches assignment if an invalid interval number is given.

**Note:**

GPU-side execution window should be updated once interval assignment finishes.
Parameters:

- `global` Global multi-GPU extraction structure describing eigenvalue extraction environment.
- `num_device` Device number whose execution window should be used.
- `window_pos` Execution window position to be assigned.
- `num_interval` Index of the interval that should be assigned to execution window.
- `start_k` Iteration number where execution is going to start.

Definition at line 85 of file gpu_eigen.cu.

### 4.3.2.2 int calc_memory_parameters (extract_global_info *global, unsigned int max_window_size)

Calculate required shared memory and maximum supported execution window size according to GPU limitations. Window size written in the `window_size` attribute of the `info` parameter.

**Note:**

Other `info` values such as `size_pq`, `size_workspace` and `window_used` are filled by this function.

Parameters:

- `global` Global multi-GPU extraction structure describing eigenvalue extraction environment.
- `max_window_size` Maximum window size specified by the user. Set to 0 to use maximum possible size.

Returns:

- **TRUE** if enough global/shared memory to solve the problem. **FALSE** otherwise.

Definition at line 207 of file gpu_eigen.cu.

### 4.3.2.3 void delete_extract_info (extract_global_info *global)

Releases all allocated memory for a multi-GPU eigenvalue extraction problem.

Parameters:

- `global` Global multi-GPU extraction structure describing eigenvalue extraction environment.

Definition at line 152 of file gpu_eigen.cu.

### 4.3.2.4 int main (int argc, char *argv[])

Definition at line 366 of file gpu_eigen.cu.

### 4.3.2.5 void parse_cmdline (int argc, char *argv[], struct gengetopt_args_info_gpu *args_info)

Parse command line commands into `args_info` structure (uses gengetopt).

Parameters:

- `argc` Number of params in command line.
**argv**  Params of command line.

**args_info**  Structure to be filled with parsed input params.

Definition at line 55 of file gpu_eigen.cu.

### 4.3.3.6 void prepare_devices (extract_global_info ∗ global)

Allocate required memory and initialize execution window of each device.

**Parameters:**

*global*  Global multi-GPU extraction structure describing eigenvalue extraction environment.

Definition at line 302 of file gpu_eigen.cu.

### 4.3.3 Variable Documentation

#### 4.3.3.1 struct gengetopt_args_info_gpu args_info  [static]

Definition at line 46 of file gpu_eigen.cu.
Kernel functions for vector operations.

#include "toeplitz.h"

Functions

• __device__ T norm (unsigned int n, const T *x, T *swork)
• __device__ T dot (unsigned int n, const T *x, const T *y, T *swork)
• __device__ T dot_reverse_y (unsigned int n, const T *x, const T *y, T *swork)
• __device__ void axpy (unsigned int n, T a, const T *x, T *y)
• __device__ void axpy_reverse_x (unsigned int n, T a, const T *x, T *y, T *swork)
• __device__ void axpxb_reverse_x (unsigned int n, T a, const T *x, T *y, T b, T *swork)

4.4.1 Detailed Description

Kernel functions for vector operations.

Author:
Leandro Graciá Gil, leagragi@inf.upv.es

Date:
18/11/08

Definition in file gpu_vector_kernel.cu.

4.4.2 Function Documentation

4.4.2.1 __device__ void axpxb_reverse_x (unsigned int n, T a, const T *x, T *y, T b, T *swork)

Performs $x = b \cdot (a \cdot \text{reverse}(x) + x)$ operation.

Parameters:

- $n$ Size of $x$ vector.
- $a$ Scalar factor.
- $x$ Input vector.
- $y$ Output vector. Can be $x$.
- $b$ Result scaling value.
- $swork$ Shared memory workspace. Requires $6 \times \text{blockSize} - 1$ floats.

Definition at line 380 of file gpu_vector_kernel.cu.
4.4.2.2 __device__ void axpy (unsigned int \( n \), T \( a \), const T * x, T * y)

Performs \( y = a \times x + y \) operation.

Parameters:
- \( n \) Size of \( x \) and \( y \) vectors.
- \( a \) Scalar factor.
- \( x \) First vector.
- \( y \) Second vector, modified with result.

Definition at line 245 of file gpu_vector_kernel.cu.

4.4.2.3 __device__ void axpy_reverse_x (unsigned int \( n \), T \( a \), const T * x, T * y, T * swork)

Performs \( y = a \times \text{reverse}(x) + y \) operation.

Parameters:
- \( n \) Size of \( x \) and \( y \) vectors.
- \( a \) Scalar factor.
- \( x \) First vector.
- \( y \) Second vector, modified with result. Cannot be \( x \).
- \( swork \) Shared memory workspace. Requires \( 5 \times \text{blockSize} - 1 \) floats.

Definition at line 278 of file gpu_vector_kernel.cu.

4.4.2.4 __device__ T dot (unsigned int \( n \), const T * x, const T * y, T * swork)

Calculates the dot product of \( x \) and \( y \) vectors.

Parameters:
- \( n \) Size of \( x \) and \( y \) vectors.
- \( x \) First vector.
- \( y \) Second vector.
- \( swork \) Shared memory workspace. Requires \( \text{blockSize} \) floats.

Returns:
- Thread 0 of the block returns the dot product result.

Definition at line 110 of file gpu_vector_kernel.cu.

4.4.2.5 __device__ T dot_reverse_y (unsigned int \( n \), const T * x, const T * y, T * swork)

Calculates the dot product of \( x \) and \( \text{reverse}(y) \) vectors.

Parameters:
- \( n \) Size of \( x \) and \( y \) vectors.
4.4 gpu_vector_kernel.cu File Reference

x First vector.
y Second vector.

swork Shared memory workspace. Requires 4 * blockSize - 1 floats.

Returns:
Thread 0 of the block returns the dot product result.

Definition at line 163 of file gpu_vector_kernel.cu.

4.4.2.6 __device__ T norm (unsigned int n, const T * x, T * swork)

Calculates the 2-norm (length) of a given vector. Should be faster than sqrt(dot(x, x)) since reads each component only once.

Parameters:

n Size of x vector.
x Input vector.

swork Shared memory workspace. Requires blockSize floats.

Returns:
Thread 0 of the block returns the dot product result.

Definition at line 40 of file gpu_vector_kernel.cu.
4.5 toeplitz.c File Reference

Toeplitz matrix related functions.

#include <math.h>
#include "toeplitz.h"

Functions

- void toeplitz_gershgorin (const T *t, unsigned int n, T *low, T *up)
- void toeplitz_mult (int n, const T *t, const T *x, T *y)
- void levinson (unsigned int n, T td, const T *t0, const T *b0, T *x)

4.5.1 Detailed Description

Toeplitz matrix related functions.

Author:
Leandro Graciá Gil, leagragi@inf.upv.es

Date:
18/11/08

Definition in file toeplitz.c.

4.5.2 Function Documentation

4.5.2.1 void levinson (unsigned int n, T td, const T *t0, const T *b0, T *x)

Solve system toeplitz(r0) * x = b0 where r0 defines a symmetric positive-definite Toeplitz matrix.

Parameters:

- \( n \) Size of \( r0, b0 \) and \( x \) vectors.
- \( td \) Main diagonal element of the symmetric Toeplitz matrix.
- \( t0 \) Rest of the symmetric Toeplitz matrix data in vector form.
- \( b0 \) Right-hand side vector data.
- \( x \) Result vector.

Note:

If matrix is not positive-definite the algorithm can still converge.

Definition at line 97 of file toeplitz.c.
4.5.2.2  void toeplitz_gershgorin (const T * t, unsigned int n, T * low, T * up)

Calculate Gershgorin disc for a given Toeplitz symmetric matrix.

Parameters:
- \( t \) Vector defining a symmetric Toeplitz matrix.
- \( n \) Size of vector \( t \).
- \( low \) Lower bound of the Gershgorin disc interval.
- \( up \) Upper bound of the Gershgorin disc interval.

Definition at line 42 of file toeplitz.c.

4.5.2.3  void toeplitz_mult (int n, const T * t, const T * x, T * y)

Calculate \( y = \text{toeplitz}(t) \ast x \) matrix-vector product where \( t \) is in vector form.

Parameters:
- \( n \) Size of \( t \) and \( x \), \( y \) vectors.
- \( t \) Data of the symmetric Toeplitz matrix in vector form.
- \( x \) Data of the input vector.
- \( y \) Data of the destination vector. Will be overwritten with the result.

Definition at line 70 of file toeplitz.c.
4.6  toeplitz.h File Reference

Main project header file.
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <f2c.h>
#include <clapack.h>
#include <gsl/gsl_cblas.h>
#include <cuda_runtime.h>

Classes
• struct interval_cpu_info_struct
• struct exec_window_entry_struct
• struct extract_device_info_struct
• struct extract_global_info_struct

Defines
• #define SINGLE
• #define blockSize 256
• #define align(x) (((x) + 255) & ~255)
• #define half_warp 16
• #define half_warp_mask 0x0F
• #define print_vector_gpu(name, v, n)
• #define T float
• #define Tsf "%f"
• #define _nrm2 cblas_snrm2
• #define _scal cblas_sscal
• #define _axpy cblas_saxpy
• #define _gemv cblas_sgemv
• #define _gemm cblas_sgemm
• #define _spmv cblas_sspmv
• #define _dot cblas_sdot
• #define _spsv sspsv_
• #define _stev ssstev_
• #define TRUE 1
• #define FALSE 0
• #define INTERVAL_NONE -1

Typedefs
• typedef struct interval_cpu_info_struct interval_cpu_info
• typedef struct exec_window_entry_struct exec_window_entry
• typedef struct extract_device_info_struct extract_device_info
• typedef struct extract_global_info_struct extract_global_info
Functions

- struct __align__ (16 *sizeof(T)) tridiagonal_entry_struct
- struct __align__ (16 *sizeof(float)) interval_gpu_info_struct
- int readInput (int *n, T **t, FILE *file)
- void print_vector (const char *name, T *v, int n)
- void print_matrix (const char *name, T *m, int cols, int rows, int lda)
- void toeplitz_gershgorin (const T *t, unsigned int n, T *low, T *up)
- void toeplitz_mult (int n, const T *t, const T *x, T *y)
- void levinson (unsigned int n, T *t, const T *t0, const T *b0, T *x)
- void sort_eigenvalues (unsigned int extracted, unsigned int n, T *el, T *ev)
- interval_cpu_info * calc_eigen_intervals (T *t, unsigned int n, T lbg, T ubg, unsigned int max_eig_-interval, unsigned int *num_intervals)
- unsigned int extract_eigenvalues (unsigned int n, unsigned int k, unsigned int max_k, unsigned int p_lda, unsigned int *converged, unsigned int *extracted, unsigned int required, T a, T b, const T *t, const T *p, const T *alpha, const T *beta, T tolerance, T *l, T *v)

Variables

- tridiagonal_entry
- interval_gpu_info

4.6.1 Detailed Description

Main project header file.

Author:

Leandro Graciá Gil, leagragi@inf.upv.es

Date:

18/11/08

Definition in file toeplitz.h.

4.6.2 Define Documentation

4.6.2.1 #define _axpy cblas_saxpy

Definition at line 71 of file toeplitz.h.

4.6.2.2 #define _dot cblas_sdot

Definition at line 75 of file toeplitz.h.

4.6.2.3 #define _gemm cblas_sgemm

Definition at line 73 of file toeplitz.h.
4.6.2.4  \#define _gemv cblas_sgemv
Definition at line 72 of file toeplitz.h.

4.6.2.5  \#define _nrm2 cblas_snrm2
Definition at line 69 of file toeplitz.h.

4.6.2.6  \#define _scal cblas_sscal
Definition at line 70 of file toeplitz.h.

4.6.2.7  \#define _spmv cblas_sspmv
Definition at line 74 of file toeplitz.h.

4.6.2.8  \#define _spsv sspsv_
Definition at line 76 of file toeplitz.h.

4.6.2.9  \#define _stev sstev_
Definition at line 77 of file toeplitz.h.

4.6.2.10 \#define align(x) (((x) + 255) & ~255)
Definition at line 55 of file toeplitz.h.

4.6.2.11 \#define blockSize 256
Definition at line 54 of file toeplitz.h.

4.6.2.12 \#define FALSE 0
Definition at line 98 of file toeplitz.h.

4.6.2.13 \#define half_warp 16
Definition at line 57 of file toeplitz.h.

4.6.2.14 \#define half_warp_mask 0x0F
Definition at line 58 of file toeplitz.h.
4.6.2.15  #define INTERVAL_NONE -1
Definition at line 102 of file toeplitz.h.

4.6.2.16  #define print_vector_gpu(name, v, n)
Value:
{
    if(tid == 0) {
        fprintf(stderr, "%s: [", name); 
        for(int _i=0; _i<n; ++_i) fprintf(stderr, " %.6f", v[_i]); 
        fprintf(stderr, "\n"]
    }
}
Definition at line 60 of file toeplitz.h.

4.6.2.17  #define SINGLE
Definition at line 45 of file toeplitz.h.

4.6.2.18  #define T float
Definition at line 67 of file toeplitz.h.

4.6.2.19  #define TRUE 1
Definition at line 94 of file toeplitz.h.

4.6.2.20  #define Tsf "%f"
Definition at line 68 of file toeplitz.h.

4.6.3  Typedef Documentation
4.6.3.1   typedef struct exec_window_entry_struct exec_window_entry
4.6.3.2   typedef struct extract_device_info_struct extract_device_info
4.6.3.3   typedef struct extract_global_info_struct extract_global_info
4.6.3.4   typedef struct interval_cpu_info_struct interval_cpu_info

4.6.4  Function Documentation
4.6.4.1   struct __align__ (16 * sizeof(float)) [read]
  < Iteration in which interval extraction was launched.
  < Flag indicating if interval has been already processed.
  < Shift value used in the interval.
Matrices of size \( n \_align \times \max_k \). Contains base vectors for symmetric and skew-symmetric Krylov subspaces.

Vector of size max(5 \( * \text{align}(n) \), 4 \( * \text{align}(n) + \max_k \)).

Struct vector of size \( \max_k \).

Definition at line 140 of file toeplitz.h.

4.6.4.2 \textbf{struct} \_\texttt{align} \((16 \times \text{sizeof}\ T)\) \[\texttt{read}\]

Definition at line 108 of file toeplitz.h.

4.6.4.3 \texttt{interval\_cpu\_info\_}\*\calc\_eigen\_intervals (T \* t, unsigned int \( n \), T lbg, T ubg, unsigned int \( \max\_eig\_interval \), unsigned int \( * \text{num\_intervals} \))

Calculates intervals containing approximately the same number of eigenvalues along a given interval. Only interval-specific information is written in the array. All other data is initialized to 0.

**Parameters:**

- \( t \) Vector defining a symmetric Toeplitz matrix whose eigenvalue intervals will be calculated.
- \( n \) Size of vector \( t \).
- \( lbg \) Lower bound of the initial interval to be subdivided (usually Gershgorin disc).
- \( ubg \) Upper bound of the initial interval to be subdivided (usually Gershgorin disc).
- \( \max\_eig\_interval \) Maximum number of eigenvalues per interval.
- \( \text{num\_intervals} \) Number of extracted intervals.

**Returns:**

Pointer to an array of size \( \text{num\_intervals} \) containing interval info. NULL in case of error.

Definition at line 105 of file eigen.c.

4.6.4.4 \texttt{unsigned\ int}\ extract\_eigen\_values (unsigned int \( n \), unsigned int \( k \), unsigned int \( \max_k \),
\texttt{unsigned\ int\ p\_lda}, \texttt{unsigned\ int}\*\ \texttt{converged}, \texttt{unsigned\ int}\*\ \texttt{extracted}, \texttt{unsigned\ int}\ \texttt{required}, \texttt{T\ a, T\ b, const\ T\*\ t, const\ T\*\ p, const\ T\*\ alpha, const\ T\*\ beta, \ T\ tolerance, T\*\ l, T\*\ v})

Check convergence and extract eigenvalues of a symmetric Toeplitz matrix given a tridiagonal shift-and-invert Lanczos approximation.

**Note:**

sigma shift value is assumed to be the halfway between \( a \) and \( b \).

**Parameters:**

- \( n \) Size of the Toeplitz matrix defined by \( t \) and number of rows of \( p \).
- \( k \) Current size of Krylov subspace and number of columns of \( p \).
- \( \max_k \) Maximum allowed size of the Krylov subspace.
- \( p\_lda \) Leading dimension of Krylov subspace vectors stored in \( p \).
4.6 toeplitz.h File Reference

**converged** Pointer to number of converged eigenvalues. Just initialize to 0 on start and let the function use it.

**extracted** Pointer to number of currently extracted eigenvalues.

**required** Number of eigenvalues that should be extracted.

**a** Start of the interval whose eigenvalues are being extracted.

**b** End of the interval whose eigenvalues are being extracted.

**t** Vector of size \( n \) defining the symmetric Toeplitz matrix.

**p** Matrix of size \( n \times k \) containing Lanczos vectors.

**alpha** Vector of size \( k \) containing the main diagonal of the symmetric tridiagonal matrix.

**beta** Vector of size \( k-1 \) containing the second diagonal of the symmetric tridiagonal matrix.

**tolerance** Tolerance threshold used to check eigenvalue convergence.

**l** Output vector where up to \( \text{required} \) eigenvalues will be stored.

**v** Output matrix where up to \( \text{required} \) eigenvectors will be stored (column order).

**Returns:**

Number of eigenvalues extracted during this call.

Definition at line 174 of file eigen.c.

4.6.4.5  void levinson (unsigned int \( n \), T \( td \), const T \* \( t0 \), const T \* \( b0 \), T \* \( x \))

Solve system \( \text{toeplitz}(r0) \ast x = b0 \) where \( r0 \) defines a symmetric positive-definite Toeplitz matrix.

**Parameters:**

\( n \) Size of \( r0, b0 \) and \( x \) vectors.

\( td \) Main diagonal element of the symmetric Toeplitz matrix.

\( t0 \) Rest of the symmetric Toeplitz matrix data in vector form.

\( b0 \) Right-hand side vector data.

\( x \) Result vector.

**Note:**

If matrix is not positive-definite the algorithm can still converge.

Definition at line 97 of file toeplitz.c.

4.6.4.6  void print_matrix (const char \* \( name \), T \* \( m \), int \( rows \), int \( cols \), int \( lda \))

Print a given matrix using standard error output.

**Parameters:**

\( name \) Name of the matrix (C-style string).

\( m \) Pointer to matrix data (column ordering).

\( rows \) Number of rows.

\( cols \) Number of columns.

\( lda \) Leading dimension of the matrix.

Definition at line 87 of file utils.c.
4.6.4.7  void print_vector (const char * name, T * v, int n)

Print a given vector using standard error output.

Parameters:

  name  Name of the vector (C-style string).
  v     Pointer to vector data.
  n     Size of the vector.

Definition at line 67 of file utils.c.

4.6.4.8  int readInput (int * n, T ** t, FILE * file)

Auxiliar function for reading input data from standard input.

Parameters:

  n     Pointer to variable that will store input data size.
  t     Pointer to variable that will store input data vector.
  file  File where contents should be read from.

Returns:

  1 if successful, 0 otherwise.

Definition at line 39 of file utils.c.

4.6.4.9  void sort_eigenvalues (unsigned int extracted, unsigned int n, T * el, T * ev)

Sort eigenvalues and eigenvectors using quicksort algorithm.

Parameters:

  extracted Number of extracted eigenvalues.
  n     Size of eigenvectors.
  el     Pointer to vector containing eigenvalues.
  ev     Pointer to matrix containing eigenvectors.

Definition at line 68 of file eigen.c.

4.6.4.10 void toeplitz_gershgorin (const T * t, unsigned int n, T * low, T * up)

Calculate Gershgorin disc for a given Toeplitz symmetric matrix.

Parameters:

  t     Vector defining a symmetric Toeplitz matrix.
  n     Size of vector t.
  low   Lower bound of the Gershgorin disc interval.
  up    Upper bound of the Gershgorin disc interval.

Definition at line 42 of file toeplitz.c.
4.6.4.11 void toeplitz_mult (int n, const T * t, const T * x, T * y)

Calculate \( y = \text{toeplitz}(t) \times x \) matrix-vector product where \( t \) is in vector form.

**Parameters:**
- \( n \) Size of \( t \) and \( x \), \( y \) vectors.
- \( t \) Data of the symmetric Toeplitz matrix in vector form.
- \( x \) Data of the input vector.
- \( y \) Data of the destination vector. Will be overwritten with the result.

Definition at line 70 of file toeplitz.c.

4.6.5 Variable Documentation

4.6.5.1 interval_gpu_info

Definition at line 148 of file toeplitz.h.

4.6.5.2 tridiagonal_entry

Definition at line 114 of file toeplitz.h.
4.7  toeplitz_kernel.cu File Reference

Kernel functions for Levinson algorithm and shift and invert 2-way Lanczos.

#include "toeplitz.h"
#include "gpu_vector_kernel.cu"

Functions

• __device__ void memcpy4 (void *dest, void *src, unsigned int size)
• __device__ void levinson (unsigned int n, T td, const T *t, T *b, T *x, T *work, T *swork)
• __device__ void reorthogonalize (unsigned int n, unsigned int k_size, const T *p, T *v, T *h, T *swork)
• __device__ void si2w (unsigned int n, T td, T *t, unsigned int min_k, unsigned int num_it, unsigned int max_k, T sigma, T *p, T *q, tridiagonal_entry *m, T *work)
• __global__ void si2w_parallel (unsigned int n, T td, T *t, unsigned int k, unsigned int inc_k, unsigned int max_k, interval_gpu_info *exec_window)

4.7.1 Detailed Description

Kernel functions for Levinson algorithm and shift and invert 2-way Lanczos.

Author:
Leandro Graciá Gil, leagragi@inf.upv.es

Date:
18/11/08

Definition in file toeplitz_kernel.cu.

4.7.2 Function Documentation

4.7.2.1 __device__ void levinson (unsigned int n, T td, const T *t, T *b, T *x, T *work, T *swork)

Solve a symmetric Toeplitz system of the form toeplitz(t) * x = b

Note:
This function requires blockSize >= half_warp.

Parameters:

n  Size of [td t], x and y vectors (size of Toeplitz matrix).

td  Diagonal value of the symmetric Toeplitz matrix.

b  Right-hand side of the system. Will be normalized by td.

x  Vector that will contain the solution of the system.
4.7 toeplitz_kernel.cu File Reference

work Work vector of size 2 * align(n).

swork Shared memory workspace. Requires 6 * blockSize - 1 floats.

Definition at line 71 of file toeplitz_kernel.cu.

4.7.2.2 __device__ void memcpy4 (void * dest, void * src, unsigned int size)

Perform a coalesced raw copy from aligned addresses with data sizes multiple of 4.

Parameters:

dest Source address. Assumed to be aligned to 16 * sizeof(float) if points to global memory.
src Destination address. Assumed to be aligned to 16 * sizeof(float) if points to global memory.
size Data size in bytes. Should be always a multiple of 4.

Definition at line 41 of file toeplitz_kernel.cu.

4.7.2.3 __device__ void reorthogonalize (unsigned int n, unsigned int k_size, const T * p, T * v, T * h, T * swork)

Reorthogonalize a vector to a Krylov subspace base.

Note:
This function requires blockSize >= half_warp.

Parameters:

n Size of v vector and number of rows of p matrix.
k_size Current size of Krylov subspace. Number of columns of p matrix.
p Pointer to a column-ordered matrix defining the base of the Krylov subspace.
v Pointer to the vector to be reorthogonalized.
h Work array of size k_size.
swork Shared memory workspace. Requires 2 * blockSize floats.

Definition at line 186 of file toeplitz_kernel.cu.

4.7.2.4 __device__ void si2w (unsigned int n, T td, T * t, unsigned int min_k, unsigned int num_it, unsigned int max_k, T sigma, T * p, T * q, tridiagonal_entry * m, T * work)

Shift and invert 2-way Lanczos routine for Toeplitz matrices. Calculates symmetric and skew-symmetric tridiagonal values, but leaves final eigenvalue calculation to CPU side.

Note:
This function requires blockSize >= half_warp.

Parameters:

n Size of the symmetric Toeplitz matrix.
**td** Main diagonal value of the symmetric Toeplitz matrix.

**t** Remaining $n-1$ values from symmetric Toeplitz matrix.

**min_k** Starting size of the Krylov subspace (starting value 0). Can be used to resume a previous extraction.

**num_it** Number of iterations to perform in the Krylov subspace.

**max_k** Maximum size of the Krylov subspace. Iterations will halt if reached.

**sigma** Shift value for centering eigenvalue extraction (depends on extraction interval).

**p** Symmetric lanczos vectors of Krylov subspace ($size \ align(n) \times max_k$).

**q** Skew-symmetric lanczos vectors of Krylov subspace ($size \ align(n) \times max_k$).

**m** Structure array containing values from symmetric and skew-symmetric tridiagonal matrices ($size max_k$).

**work** Workspace array of size $>= \ max(5 \ * \ align(n), 4 \ * \ align(n) + \ max_k)$.

Definition at line 266 of file toeplitz_kernel.cu.

### 4.7.2.5 __global__ void si2w_parallel (unsigned int n, T td, T *t, unsigned int k, unsigned int inc_k, unsigned int max_k, interval_gpu_info *exec_window)

Entry point for parallel shift-and-invert 2-way Lanczos evaluation.

**Parameters:**

- **n** Size of the symmetric Toeplitz matrix.
- **td** Main diagonal value of the symmetric Toeplitz matrix.
- **t** Remaining $n-1$ values from symmetric Toeplitz matrix.
- **k** Current value of subspace size iterator. Real subspace size will be calculated using interval data.
- **inc_k** Number of iterations to perform in the Krylov subspace.
- **max_k** Maximum size of the Krylov subspace. Iterations will halt if reached.
- **exec_window** Pointer to execution window containing GPU-side descriptors of the intervals to be calculated.

Definition at line 397 of file toeplitz_kernel.cu.
4.8 utils.c File Reference

Utility functions for eigenvalue extraction tool.
#include "toeplitz.h"

Functions

- int readInput (int *n, T **t, FILE *file)
- void print_vector (const char *name, T *v, int n)
- void print_matrix (const char *name, T *m, int rows, int cols, int lda)

4.8.1 Detailed Description

Utility functions for eigenvalue extraction tool.

Author:
Leandro Graciá Gil, leagragi@inf.upv.es

Date:
18/11/08

Definition in file utils.c.

4.8.2 Function Documentation

4.8.2.1 void print_matrix (const char * name, T * m, int rows, int cols, int lda)

Print a given matrix using standard error output.

Parameters:

- name Name of the matrix (C-style string).
- m Pointer to matrix data (column ordering).
- rows Number of rows.
- cols Number of columns.
- lda Leading dimension of the matrix.

Definition at line 87 of file utils.c.

4.8.2.2 void print_vector (const char * name, T * v, int n)

Print a given vector using standard error output.

Parameters:

- name Name of the vector (C-style string).
- v Pointer to vector data.
- n Size of the vector.

Definition at line 67 of file utils.c.
4.8.2.3 int readInput (int *n, T **t, FILE *file)

Auxiliar function for reading input data from standard input.

Parameters:

n Pointer to variable that will store input data size.
t Pointer to variable that will store input data vector.
file File where contents should be read from.

Returns:

1 if successful, 0 otherwise.

Definition at line 39 of file utils.c.
Index

__align__
  toeplitz.h, 37, 38
_axpy
  toeplitz.h, 35
__dot
  toeplitz.h, 35
__gemm
  toeplitz.h, 35
__gemv
  toeplitz.h, 35
__nrm2
  toeplitz.h, 36
__scal
  toeplitz.h, 36
__spmv
  toeplitz.h, 36
__spsv
  toeplitz.h, 36
__stev
  toeplitz.h, 36
align
  toeplitz.h, 36
alpha
  interval_cpu_info_struct, 15
__args_info
  cpu_eigen.c, 22
  gpu_eigen.cu, 28
assign_interval
  gpu_eigen.cu, 26
axpxb_reverse_x
  gpu_vector_kernel.cu, 29
apx
  gpu_vector_kernel.cu, 29
apx_reverse_x
  gpu_vector_kernel.cu, 30
beta
  interval_cpu_info_struct, 15
blockSize
  toeplitz.h, 36
calc_eigen_intervals
  eigen.c, 24
  toeplitz.h, 38
calc_intervals
  eigen.c, 24
calc_memory_parameters
gpu_eigen.cu, 27
converged_a
  interval_cpu_info_struct, 16
converged_s
  interval_cpu_info_struct, 16
copy_stream
  exec_window_entry_struct, 5
cpu
  extract_global_info_struct, 12
cpu_alpha
  extract_device_info_struct, 8
cpu_beta
  extract_device_info_struct, 8
cpu_delta
  extract_device_info_struct, 8
cpu_eigen.c, 21
  args_info, 22
  main, 22
  parse_cmdline, 22
  si2w_cpu, 22
cpu_gamma
  extract_device_info_struct, 8
cpu_m
  extract_device_info_struct, 8
cpu_p
  extract_device_info_struct, 8
cpu_q
  extract_device_info_struct, 8
delete_extract_info
  gpu_eigen.cu, 27
delta
  interval_cpu_info_struct, 16
device
  extract_global_info_struct, 12
device_prop
  extract_device_info_struct, 7
dot
  gpu_vector_kernel.cu, 30
dot_reverse_y
  gpu_vector_kernel.cu, 30
eigen.c, 23
calc_eigen_intervals, 24
calc_intervals, 24
extract_eigenvalues, 24
interval_node, 24
sort_aux_struct, 24
sort_eigenvalues, 25
sort_eigenvalues_aux, 25
eigen_left
interval_cpu_info_struct, 16
eigs
interval_node_struct, 18
el
sort_aux_struct, 19
evec_stream
extract_device_info_struct, 7
evec_window
extract_device_info_struct, 7
evec_window_entry
toeplitz.h, 37
evec_window_entryStruct, 5
copy_stream, 5
interval_index, 5
evec_window_gpu
extract_device_info_struct, 7
extract_device_info
toeplitz.h, 37
extract_device_info_struct, 6
cpu_alpha, 8
cpu_beta, 8
cpu_delta, 8
cpu_gamma, 8
cpu_m, 8
cpu_p, 8
cpu_q, 8
device_prop, 7
evec_stream, 7
evec_window, 7
evec_window_gpu, 7
gpu_m, 8
gpu_p, 7
gpu_q, 8
gpu_t, 7
gpu_workspace, 8
window_size, 7
window_used, 7
extract_eigenvalues
eigen.c, 24
toeplitz.h, 38
extract_global_info
toeplitz.h, 37
extract_global_info_struct, 10
cpu, 12
device, 12
global_window_used, 12
gpu, 12
inc_k, 11
max_eig_interval, 11
max_k, 11
max_retry_count, 11
n, 11
next_interval, 12
num_devices, 12
num_intervals, 11
size_pq, 12
size_smem, 13
size_toeplitz_data, 13
size_workspace, 13
t, 12
tolerance, 12
extracted_a
interval_cpu_info_struct, 16
extracted_s
interval_cpu_info_struct, 16
FALSE
toeplitz.h, 36
gamma
interval_cpu_info_struct, 16
global_window_used
extract_global_info_struct, 12
gpu
extract_global_info_struct, 12
gpu.eigen.cu, 26
args_info, 28
assign_interval, 26
calc_memory_parameters, 27
delete_extract_info, 27
main, 27
parse_cmdline, 27
prepare_devices, 28
gpu_m
extract_device_info_struct, 8
gpu_p
extract_device_info_struct, 7
gpu_q
extract_device_info_struct, 8
gpu_t
extract_device_info_struct, 7
gpu_vector_kernel.cu, 29
axpxb_reverse_x, 29
axpy, 29
axpy_reverse_x, 30
dot, 30
dot_reverse_y, 30
norm, 31
gpu_workspace
extract_device_info_struct, 8
half_warp
toeplitz.h, 36
half_warp_mask
toeplitz.h, 36
inc_k
extract_global_info_struct, 11
index
sort_aux_struct, 19
interval_cpu_info
toeplitz.h, 37
interval_cpu_info_struct, 14
alpha, 15
beta, 15
converged_a, 16
converged_s, 16
delta, 16
eigen_left, 16
extracted_a, 16
extracted_s, 16
gamma, 16
la, 15
lb, 15
ls, 15
m, 16
num_eigs, 15
p, 15
q, 15
retry_count, 16
ub, 15
va, 15
vs, 15
interval_gpu_info
toeplitz.h, 41
interval_index
exec_window_entry_struct, 5
interval_node
eigen.c, 24
interval_node_struct, 18
eigs, 18
lb, 18
next, 18
post_eigs, 18
prev_eigs, 18
ub, 18
INTERVAL_NONE
toeplitz.h, 36
la
interval_cpu_info_struct, 15
lb
interval_cpu_info_struct, 15
interval_node_struct, 18
levinson
toeplitz.c, 32
toeplitz.h, 39
toeplitz_kernel.cu, 42
ls
interval_cpu_info_struct, 15
m
interval_cpu_info_struct, 16
main
cpu_eigen.c, 22
gpu_eigen.cu, 27
max_eig_interval
extract_global_info_struct, 11
max_k
extract_global_info_struct, 11
max_retry_count
extract_global_info_struct, 11
memcpy4
toeplitz_kernel.cu, 43
n
extract_global_info_struct, 11
next
interval_node_struct, 18
next_interval
extract_global_info_struct, 12
norm
gpu_vector_kernel.cu, 31
num_devices
extract_global_info_struct, 12
num_eigs
interval_cpu_info_struct, 15
num_intervals
extract_global_info_struct, 11
p
interval_cpu_info_struct, 15
parse_cmdline
cpu_eigen.c, 22
gpu_eigen.cu, 27
post_eigs
interval_node_struct, 18
prepare_devices
gpu_eigen.cu, 28
prev_eigs
interval_node_struct, 18
print_matrix
toeplitz.h, 39
utils.c, 45
print_vector
toeplitz.h, 39
utils.c, 45
print_vector_gpu
    toeplitz.h, 37

q
    interval_cpu_info_struct, 15

readInput
    toeplitz.h, 40
    utils.c, 45

reorthogonalize
    toeplitz_kernel.cu, 43

retry_count
    interval_cpu_info_struct, 16

si2w
    toeplitz_kernel.cu, 43
si2w_cpu
    cpu_eigen.c, 22
si2w_parallel
    toeplitz_kernel.cu, 44

SINGLE
    toeplitz.h, 37

size_pq
    extract_global_info_struct, 12

size_smem
    extract_global_info_struct, 13

size_toeplitz_data
    extract_global_info_struct, 13

size_workspace
    extract_global_info_struct, 13

sort_aux_struct
    eigen.c, 24

sort_aux_struct_, 19
    el, 19
    index, 19

sort_eigenvalues
    eigen.c, 25
    toeplitz.h, 40

sort_eigenvalues_aux
    eigen.c, 25

T
    toeplitz.h, 37

t
    extract_global_info_struct, 12

    toeplitz.c, 32
    levinson, 32
    toeplitz_gershgorin, 32
    toeplitz_mult, 33

    toeplitz.h, 34
    __align__, 37, 38
    __axpy__, 35
    __dot__, 35
    __gemm__, 35

    _gemv, 35
    _nrm2, 36
    _scal, 36
    _spmv, 36
    _spsv, 36
    _stev, 36
    align, 36
    blockSize, 36
    calc_eigen_intervals, 38
    exec_window_entry, 37
    extract_device_info, 37
    extract_eigenvalues, 38
    extract_global_info, 37
    FALSE, 36
    half_warp, 36
    half_warp_mask, 36
    interval_cpu_info, 37
    interval_gpu_info, 41
    INTERVAL_NONE, 36
    levinson, 39
    print_matrix, 39
    print_vector, 39
    print_vector_gpu, 37
    readInput, 40
    SINGLE, 37
    sort_eigenvalues, 40
    T, 37
    toeplitz_gershgorin, 40
    toeplitz_mult, 40
    tridiagonal_entry, 41
    TRUE, 37
    Tsf, 37

    toeplitz_gershgorin
    toeplitz.c, 32
    toeplitz.h, 40

    toeplitz_kernel.cu, 42
    levinson, 42
    memcpy4, 43
    reorthogonalize, 43
    si2w, 43
    si2w_parallel, 44

    toeplitz_mult
    toeplitz.c, 33
    toeplitz.h, 40

tolerance
    extract_global_info_struct, 12

    tridiagonal_entry
    toeplitz.h, 41
    TRUE
    toeplitz.h, 37
    Tsf
    toeplitz.h, 37

    ub
INDEX

interval_cpu_info_struct, 15
interval_node_struct, 18
utils.c, 45
    print_matrix, 45
    print_vector, 45
    readInput, 45

va
    interval_cpu_info_struct, 15
vs
    interval_cpu_info_struct, 15

window_size
    extract_device_info_struct, 7
window_used
    extract_device_info_struct, 7