

# MMTK: MATLAB Wrappers for the Mimetic Methods Toolkit

Generated by Doxygen 1.8.6

Thu Sep 10 2015 15:44:26



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	MTK Concerns . . . . .	1
1.2	MTK Flavors . . . . .	1
1.3	Contact, Support and Credits . . . . .	2
1.4	Acknowledgements and Contributions . . . . .	2
<b>2</b>	<b>Programming Tools</b>	<b>3</b>
<b>3</b>	<b>Licensing and Modifications</b>	<b>5</b>
<b>4</b>	<b>Read Me File and Installation Instructions</b>	<b>7</b>
<b>5</b>	<b>Tests and Test Architectures</b>	<b>15</b>
<b>6</b>	<b>Examples</b>	<b>17</b>
<b>7</b>	<b>User Manual, References and Theory</b>	<b>19</b>
<b>8</b>	<b>File Index</b>	<b>21</b>
8.1	File List . . . . .	21
<b>9</b>	<b>File Documentation</b>	<b>23</b>
9.1	examples/Div1D.m File Reference . . . . .	23
9.1.1	Variable Documentation . . . . .	23
9.1.1.1	all . . . . .	23
9.1.1.2	clc . . . . .	23
9.1.1.3	shortG . . . . .	23
9.2	Div1D.m . . . . .	23
9.3	include/mmtk.h File Reference . . . . .	24
9.3.1	Detailed Description . . . . .	24
9.4	mmtk.h . . . . .	24
9.5	Makefile.inc File Reference . . . . .	25

9.6	Makefile.inc	25
9.7	mexsrc/MMTKDiv1D.cc File Reference	27
9.7.1	Detailed Description	27
9.8	MMTKDiv1D.cc	27
9.9	mexsrc/MMTKDiv1D.m File Reference	31
9.10	MMTKDiv1D.m	31
9.11	README.md File Reference	31
9.12	README.md	31

<b>Index</b>	<b>35</b>
--------------	-----------

# Chapter 1

## Introduction

We define numerical methods that are based on discretizations preserving the properties of their continuum counterparts to be **mimetic**.

The **Mimetic Methods Toolkit (MTK)** is a C++ library for mimetic numerical methods. It is a set of classes for **mimetic quadratures**, **mimetic interpolation**, and **mimetic discretization methods** for the numerical solution of ordinary and partial differential equations.

### 1.1 MTK Concerns

Since collaborative development efforts are definitely important in achieving the level of generality we intend the library to possess, we have divided the library's source code according to the designated purpose the classes possess within the library. These divisions (or concerns) are grouped by layers, and are hierarchically related by the dependence they have among them.

One concern is said to depend on another one, if the classes the first concern includes, rely on the classes the second concern includes.

In order of dependence these are:

1. Roots.
2. Enumerations.
3. Tools.
4. Data Structures.
5. Numerical Methods.
6. Grids.
7. Mimetic Operators.

### 1.2 MTK Flavors

The MTK collection of wrappers is:

1. MMTK: MATLAB wrappers collection for MTK; intended for sequential computations.

Others are being designed and developed.

### 1.3 Contact, Support and Credits

The MTK is developed by researchers and adjuncts to the Computational Science Research Center (CSRC) at San Diego State University (SDSU).

Developers are members of:

1. Mimetic Numerical Methods Research and Development Group.
2. Computational Geoscience Research and Development Group.
3. Ocean Modeling Research and Development Group.

Currently the developers are:

1. **Eduardo J. Sanchez, Ph.D. - esanchez at mail dot sdsu dot edu - ejspeiro**
2. Jose E. Castillo, Ph.D. - jcastillo at mail dot sdsu dot edu
3. Guillermo F. Miranda, Ph.D. - unigrav at hotmail dot com
4. Christopher P. Paolini, Ph.D. - paolini at engineering dot sdsu dot edu
5. Angel Boada.
6. Johnny Corbino.
7. Raul Vargas-Navarro.

### 1.4 Acknowledgements and Contributions

The authors would like to acknowledge valuable advising, contributions and feedback, from research personnel at the Computational Science Research Center at San Diego State University, which were vital to the fruition of this work. Specifically, our thanks go to (alphabetical order):

1. Mohammad Abouali, Ph.D.
2. Dany De Cecchis, Ph.D.
3. Julia Rossi.

## Chapter 2

# Programming Tools

The development of MTK has been made possible through the use of the following applications:

1. Editor: Kate - KDE Advanced Text Editor. Version 3.13.3. Using KDE Development Platform 4.13.3 (C) 2000-2005 The Kate Authors.
2. Compiler: gcc version 4.4.5 (Ubuntu/Linaro 4.4.4-14ubuntu5). Copyright (C) 2013 Free Software Foundation, Inc.
3. Debugger: GNU gdb (Ubuntu 7.7.1-0ubuntu5~14.04.2) 7.7.1. Copyright (C) 2014 Free Software Foundation, Inc.





## Chapter 3

# Licensing and Modifications

Copyright (C) 2015, Computational Science Research Center, San Diego State University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Modifications to source code should be reported to: [esanchez@mail.sdsu.edu](mailto:esanchez@mail.sdsu.edu) and a copy of the modified files should be reported once modifications are completed. Documentation related to said modifications should be included.
2. Redistributions of source code must be done through direct downloads from the project's GitHub page: <http://www.csrc.sdsu.edu/mtk>
3. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
4. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
5. Usage of the binary form on proprietary applications shall require explicit prior written permission from the the copyright holders.
6. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The copyright holders provide no reassurances that the source code provided does not infringe any patent, copyright, or any other intellectual property rights of third parties. The copyright holders disclaim any liability to any recipient for claims brought against recipient by any third party for infringement of that parties intellectual property rights.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## Chapter 4

# Read Me File and Installation Instructions

### README File for the Mimetic Methods Toolkit (MTK)

By: **Eduardo J. Sanchez, Ph.D.** - `esanchez at mail dot sdsu dot edu`

#### 1. Description

We define numerical methods that are based on discretizations preserving the properties of their continuum counterparts to be **mimetic**.

The **Mimetic Methods Toolkit (MTK)** is a C++ library for mimetic numerical methods. It is arranged as a set of classes for **mimetic quadratures**, **mimetic interpolation**, and **mimetic discretization** methods for the numerical solution of ordinary and partial differential equations.

This collection of **MATLAB Wrappers for the MTK (MMTK)** allows developers to invoke the MTK from a MATLAB environment.

---

#### 2. Dependencies

This README assumes all of these dependencies are installed in the following folder:

```
```\n/home/ejspeiro/Libraries/\n```
```

In this version, the MTK optionally uses ATLAS-optimized BLAS and LAPACK routines for the internal computation on some of the layers. However, ATLAS requires both BLAS and LAPACK in order to create their optimized distributions. Therefore, the following dependencies tree arises:

#### For Linux and OS X:

1. MATLAB R2014a or greater - Available from: <http://www.mathworks.com/>
1. MTK - Available from: <http://csrc.sdsu.edu/mtk/>

### 3. Installation

The following instructions assume MATLAB R2014a or greater.

You have two options, either follow the instructions given, **expert install**, or use the provided **patch file**, **naive install**.

#### EXPERT PART 1. CONFIGURATION OF THE MEX COMPILER.

From your MATLAB prompt, type:

```
```\n\nmex -setup C++
```

MEX configured to use 'g++' for C++ language compilation.  
Warning: The MATLAB C and Fortran API has changed to support MATLAB variables with more than  $2^{32}-1$  elements. In the near future you will be required to update your code to utilize the new API. You can find more information about this at:

---

[http://www.mathworks.com/help/matlab/matlab\\_external/upgrading-mex-files-to-use-64-bit-api.html](http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html).

'''

## EXPERT PART 2. CONFIGURATION OF THE MEX COMPILER SETUP FILE.

The previous step creates the following file:

```
'''
/home/ejspeiro/.matlab/R2014a/mex_C++_glnxa64.xml
'''
```

The purpose of this section is to configure the MATLAB R2014a Mex compiler so that it can work with the latest C++ standard (C++11).

Please execute the following changes on the aforementioned file (line numbers may differ):

Lines 26 and 27:

```
'''
CMDLINE1="$CXX -std=c++11 -c $DEFINES $INCLUDE $CXXFLAGS $OPTIM $SRC -o $OBJ"
CMDLINE2="$LDXX -std=c++11 $LD_FLAGS $LDTYPE $LINKOPTIM $LINKEXPORT $OBJ $CXXLIBS $LINKLIBS -o $EXE"
'''
```

Line 33:

```
'''
CXXFLAGS="-std=c++11 -ansi -fexceptions -fPIC -fno-omit-frame-pointer -pthread"
'''
```

Line 35 and 26:

```
'''
CXXOPTIMFLAGS="-std=c++11 -O -DNDEBUG"
CXXDEBUGFLAGS="-std=c++11 -g"
'''
```

Line 38:

```
```\nLDXX="gfortran"\n```
```

Line 59:

```
```\nCXXFLAGS="-std=c++11 -ansi -pthread"\n```
```

#### NAIVE PART 1. CONFIGURATION OF THE MEX COMPILER.

From your MATLAB prompt, type:

```
```
```

```
mex -setup C++
```

MEX configured to use 'g++' for C++ language compilation.  
Warning: The MATLAB C and Fortran API has changed to support MATLAB  
variables with more than  $2^{32}-1$  elements. In the near future  
you will be required to update your code to utilize the  
new API. You can find more information about this at:

[http://www.mathworks.com/help/matlab/matlab\\_external/upgrading-mex-files-to-use-64-bit-api.html](http://www.mathworks.com/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html).

```
```
```

---

## NAIVE PART 2. CONFIGURATION OF THE MEX COMPILER SETUP FILE.

```

'''
cd $HOME/.matlab/R2014a
chmod +w mex_C++_glnxa64.xml
patch < mex_C++_glnxa64.patch
chmod -w mex_C++_glnxa64.xml
'''
Exit terminal, and restart MATLAB. You can use C++11 to create MEX files now!

```

## PART 3: CONFIGURATION OF THE MAKEFILE.

The following steps are required to build and test the MTK. Please use the accompanying [Makefile.inc](#) file, which should provide a solid template to start with. The following command provides help on the options for make:

```
'''
```

### \$ make help

Makefile for the MMTK.

Options are:

- all: builds the library, the tests, and examples.
- gendoc: generates the documentation for the library.
- - clean: cleans ALL the generated files.

```
'''
```

## PART 4. BUILD THE MMTK.

From your shell, at the base folder of the MMTK, just type:

```
```\nmake\n```
```

If successful you'll read:

```
```\n----- Library created! Check in /home/ejspeiro/Dropbox/MTK/lib\n```
```

## 4. Frequently Asked Questions

Q: Why haven't you guys implemented GBS to build the library?

A: I'm on it as we speak! ;)

Q: When will the other flavors be ready?

A: Soon! I'm working on getting help on developing those.

Q: Is there any main reference when it comes to the theory on Mimetic Methods?

A: Yes! Check: <http://www.csrc.sdsu.edu/mimetic-book>

Q: Do I need to generate the documentation myself?

A: You can if you want to... but if you DO NOT want to, just go to our website.

## 5. Contact, Support, and Credits

The MTK is developed by researchers and adjuncts to the  
[Computational Science Research Center \(CSRC\)](#)  
at [San Diego State University \(SDSU\)](#).

Developers are members of:

1. Mimetic Numerical Methods Research and Development Group.
2. Computational Geoscience Research and Development Group.
3. Ocean Modeling Research and Development Group.

Currently the developers are:

**Eduardo J. Sanchez, Ph.D. - [esanchez at mail dot sdsu dot edu](mailto:esanchez@mail.sdsu.edu) -**

- Jose E. Castillo, Ph.D. - [jcastillo at mail dot sdsu dot edu](mailto:jcastillo@mail.sdsu.edu)



- Guillermo F. Miranda, Ph.D. - unigrav at hotmail dot com
- Christopher P. Paolini, Ph.D. - paolini at engineering dot sdsu dot edu
- Angel Boada.
- Johnny Corbino.
- Raul Vargas-Navarro.

Finally, please feel free to contact me with suggestions or corrections:

**Eduardo J. Sanchez, Ph.D. - esanchez at mail dot sdsu dot edu -**

Thanks and happy coding!



## Chapter 5

# Tests and Test Architectures

Tests are given in the `files list` section. They are provided in the `/tests/` folder within the distributed software.

In this page we intend to make a summary of all of the architectures in where the MTK has been tested. The MTK is intended to be as portable as possible throughout architectures. The following architectures have provided flawless installations of the API and correct execution of the examples:

1. Linux 3.2.0-23-generic-pae #36-Ubuntu SMP i386 GNU/Linux  
Intel(R) Pentium(R) M processor 1.73GHz 2048 KB of cache and stepping of 8  
gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5)

Further architectures will be tested!



## Chapter 6

# Examples

Examples are given in the `files list` section. They are provided in the `/examples/` folder within the distributed software.



## Chapter 7

# User Manual, References and Theory

The main source of references for this work can be found in:

<http://www.csrc.sdsu.edu/mimetic-book/>

However, a .PDF copy of this manual can be found [here](#).





## Chapter 8

# File Index

### 8.1 File List

Here is a list of all files with brief descriptions:

<a href="#">Makefile.inc</a>	25
examples/ <a href="#">Div1D.m</a>	23
include/ <a href="#">mmtk.h</a>	
Includes the entire API	24
mexsrc/ <a href="#">MMTKDiv1D.cc</a>	
MEX file for the Div1D class	27
mexsrc/ <a href="#">MMTKDiv1D.m</a>	31



## Chapter 9

# File Documentation

### 9.1 examples/Div1D.m File Reference

#### Variables

- format [shortG](#)
- close [all](#)
- [clc](#)

#### 9.1.1 Variable Documentation

##### 9.1.1.1 clear all

Definition at line [22](#) of file [Div1D.m](#).

##### 9.1.1.2 clc

Definition at line [24](#) of file [Div1D.m](#).

##### 9.1.1.3 format shortG

Definition at line [21](#) of file [Div1D.m](#).

### 9.2 Div1D.m

```
00001 % FUNCTION_NAME - Div1D
00002 % Creation of different high-order mimetic divergence operators with the MMTK.
00003 %
00004 % Other m-files required: None.
00005 % Subfunctions: The MMTK.
00006 % MAT-files required: None.
00007 %
00008 % See also: MMTKDiv1D.
00009 %
00010 % Author: Eduardo Sanchez, Ph.D.
00011 % Computational Science Research Center.
00012 % 5500 Campanile Drive, San Diego State University.
00013 % Email: esanchez at mail dot sdsu dot edu
```

```

00014 % Website: http://www-rohan.sdsu.edu/~sanche94/
00015 % 2015-09-10 14:18:35
00016
00017 %----- BEGIN CODE -----
00018
00019 addpath(' ../mexsrc ');
00020
00021 format shortG;
00022 close all;
00023 clear all;
00024 clc;
00025
00026 MMTKDiv1D(2)
00027
00028 %----- END OF CODE -----

```

## 9.3 include/mmtk.h File Reference

Includes the entire API.

### 9.3.1 Detailed Description

This file contains the related documentation

#### Author

: Eduardo J. Sanchez: esanchez at mail dot sdsu dot edu

#### Warning

This file contains no logic for the MMTK. It has been created just for documentation purposes.

Definition in file [mmtk.h](#).

## 9.4 mmtk.h

```

00001
00013 /*
00014 Copyright (C) 2015, Computational Science Research Center, San Diego State
00015 University. All rights reserved.
00016
00017 Redistribution and use in source and binary forms, with or without modification,
00018 are permitted provided that the following conditions are met:
00019
00020 1. Modifications to source code should be reported to: esanchez@mail.sdsu.edu
00021 and a copy of the modified files should be reported once modifications are
00022 completed. Documentation related to said modifications should be included.
00023
00024 2. Redistributions of source code must be done through direct
00025 downloads from the project's GitHub page: http://www.csrc.sdsu.edu/mtk
00026
00027 3. Redistributions of source code must retain the above copyright notice, this
00028 list of conditions and the following disclaimer.
00029
00030 4. Redistributions in binary form must reproduce the above copyright notice,
00031 this list of conditions and the following disclaimer in the documentation and/or
00032 other materials provided with the distribution.
00033
00034 5. Usage of the binary form on proprietary applications shall require explicit
00035 prior written permission from the the copyright holders.
00036
00037 6. Neither the name of the copyright holder nor the names of its contributors
00038 may be used to endorse or promote products derived from this software without
00039 specific prior written permission.

```

```

00040
00041 The copyright holders provide no reassurances that the source code provided does
00042 not infringe any patent, copyright, or any other intellectual property rights of
00043 third parties. The copyright holders disclaim any liability to any recipient for
00044 claims brought against recipient by any third party for infringement of that
00045 parties intellectual property rights.
00046
00047 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
00048 ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
00049 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
00050 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
00051 ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
00052 (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
00053 LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
00054 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00055 (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
00056 SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00057 */

```

## 9.5 Makefile.inc File Reference

## 9.6 Makefile.inc

```

00001 # Makefile setup file for MMTK.
00002
00003 SHELL := /bin/bash
00004
00005 # Please set the following variables up:
00006
00007 # 2. Absolute path to base directory of the MTK... where is the MTK?
00008 # _____
00009
00010 BASE = /home/ejspeiro/Dropbox/MTK
00011
00012 # 2. The machine (platform) identifier and required precision.
00013 # _____
00014
00015 # Options are:
00016 # - LINUX: A LINUX box installation.
00017 # - OSX: Uses OS X optimized solvers.
00018
00019 PLAT = LINUX
00020
00021 # Options are:
00022 # - SINGLE: Use 4 B floating point numbers.
00023 # - DOUBLE: Use 8 B floating point numbers.
00024
00025 PRECISION = DOUBLE
00026
00027 # 3. Optimized solvers and operations by means of ATLAS in Linux?
00028 # _____
00029
00030 # If you have selected OSX in step 1, then you don't need to worry about this.
00031
00032 # Options are ON xor OFF:
00033
00034 ATL_OPT = OFF
00035
00036 # 4. Paths to dependencies (header files for compiling).
00037 # _____
00038
00039 # GLPK include path (soon to go):
00040
00041 GLPK_INC = $(HOME)/Libraries/glpk-4.55/include
00042
00043 # Linux: If ATLAS optimization is ON, users should only provide the path to
00044 # ATLAS:
00045
00046 ATLAS_INC = $(HOME)/Libraries/ATLAS_3.8.4-CORE/include
00047
00048 # OS X: Do nothing.
00049
00050 # 5. Paths to dependencies (archive files for (static) linking).
00051 # _____
00052

```

```

00053 # GLPK linking path (soon to go):
00054
00055 GLPK_LIB = $(HOME)/Libraries/glpk-4.55/lib/libglpk.a
00056
00057 # If optimization is OFF, then provide the paths for:
00058
00059 BLAS_LIB = $(HOME)/Libraries/BLAS/libblas.a
00060 LAPACK_LIB = $(HOME)/Libraries/lapack-3.4.1/liblapack.a
00061
00062 # WARNING: Vendor libraries should be used whenever they are available.
00063
00064 # However, if optimization is ON, please provide the path the ATLAS' archive:
00065
00066 ATLAS_LIB = $(HOME)/Libraries/ATLAS_3.8.4-CORE/ATLAS_3.8.4-BUILD-Citadel/lib
00067
00068 # 6. Compiler and its flags.
00069 # _____
00070
00071 # Debug Level. Options are:
00072 # 0. NO debug at all NOR any run-time checks... be cautious!
00073 # 1. Verbose (execution messages) AND run-time checks.
00074 # 2. Level 1 plus intermediate scalar-valued results.
00075 # 3. Level 2 plus intermediate array-valued results.
00076
00077 DEBUG_LEVEL = 3
00078
00079 # Flags recommended for release code:
00080
00081 CCFLAGS = -O2
00082
00083 # Flags recommended for debugging code:
00084
00085 CCFLAGS =
00086
00087 # 7. Archiver, its flags, and ranlib:
00088 # _____
00089
00090 ARCH = ar
00091 ARCHFLAGS = cr
00092
00093 # If your system does not have "ranlib" then set: "RANLIB = echo":
00094
00095 RANLIB = echo
00096
00097 # But, if possible:
00098
00099 RANLIB = ranlib
00100
00101 # 8. Valgrind's memcheck options:
00102 # _____
00103
00104 MEMCHECK_OPTS = -v --tool=memcheck --leak-check=full --show-leak-kinds=all \
00105 --track-origins=yes --freelist-vol=20000000
00106
00107 # Done!
00108
00109 # _____
00110 # _____
00111 # _____
00112
00113 # MMTK-related.
00114 # _____
00115
00116 SRC = $(BASE)/src
00117 INCLUDE = $(BASE)/include
00118 LIB = $(BASE)/lib
00119 MTK_LIB = $(LIB)/libmtk.a
00120 TESTS = $(BASE)/tests
00121 EXAMPLES = $(BASE)/examples
00122
00123 # Compiling-related.
00124 # _____
00125
00126 CC = mex
00127
00128 CCFLAGS += -largeArrayDims -g
00129
00130 ifeq ($(PRECISION),DOUBLE)
00131 CCFLAGS += -DMTK_PRECISION_DOUBLE
00132 else
00133 CCFLAGS += -DMTK_PRECISION_SINGLE

```

```

00134 endif
00135
00136 CCFLAGS += -DMTK_DEBUG_LEVEL=$(DEBUG_LEVEL) -I$(INCLUDE)
00137
00138 # Only the GLPK is included because the other dependencies are coded in Fortran.
00139
00140 ifeq ($(ATL_OPT),ON)
00141     CCFLAGS += -I$(GLPK_INC) $(ATLAS_INC)
00142 else
00143     CCFLAGS += -I$(GLPK_INC)
00144 endif
00145
00146 # Linking-related.
00147 # -----
00148
00149 NOOPT_LIBS = $(LAPACK_LIB) $(GLPK_LIB) $(BLAS_LIB) -lstdc++ -lgfortran
00150
00151 OPT_LIBS = -L$(ATLAS_LIB) -latlas -llapack -lblas -lm -latlas \
00152 -lstdc++ -lgfortran
00153
00154 LIBS = $(MTK_LIB)
00155
00156 ifeq ($(PLAT),OSX)
00157     LIBS += -framework Accelerate $(GLPK_LIB)
00158 else
00159     ifeq ($(ATL_OPT),ON)
00160         LIBS += $(OPT_LIBS)
00161     else
00162         LIBS += $(NOOPT_LIBS)
00163     endif
00164 endif
00165
00166 # Documentation-related.
00167 # -----
00168
00169 DOCGEN = doxygen
00170 DOCFilename = doc_config.dxcf
00171 DOC = $(HOME)/Dropbox/MMTK/doc
00172 DOCFILE = $(HOME)/Dropbox/MMTK/$(DOCFilename)

```

## 9.7 mexsrc/MMTKDiv1D.cc File Reference

MEX file for the Div1D class.

### 9.7.1 Detailed Description

#### Author

: Eduardo J. Sanchez (ejspeiro) - esanchez at mail dot sdsu dot edu

Definition in file [MMTKDiv1D.cc](#).

## 9.8 MMTKDiv1D.cc

```

00001
00008 /*
00009 Copyright (C) 2015, Computational Science Research Center, San Diego State
00010 University. All rights reserved.
00011
00012 Redistribution and use in source and binary forms, with or without modification,
00013 are permitted provided that the following conditions are met:
00014
00015 1. Modifications to source code should be reported to: esanchez@mail.sdsu.edu
00016 and a copy of the modified files should be reported once modifications are
00017 completed. Documentation related to said modifications should be included.
00018
00019 2. Redistributions of source code must be done through direct
00020 downloads from the project's GitHub page: http://www.csrc.sdsu.edu/mtk

```

```

00021
00022 3. Redistributions of source code must retain the above copyright notice, this
00023 list of conditions and the following disclaimer.
00024
00025 4. Redistributions in binary form must reproduce the above copyright notice,
00026 this list of conditions and the following disclaimer in the documentation and/or
00027 other materials provided with the distribution.
00028
00029 5. Usage of the binary form on proprietary applications shall require explicit
00030 prior written permission from the the copyright holders.
00031
00032 6. Neither the name of the copyright holder nor the names of its contributors
00033 may be used to endorse or promote products derived from this software without
00034 specific prior written permission.
00035
00036 The copyright holders provide no reassurances that the source code provided does
00037 not infringe any patent, copyright, or any other intellectual property rights of
00038 third parties. The copyright holders disclaim any liability to any recipient for
00039 claims brought against recipient by any third party for infringement of that
00040 parties intellectual property rights.
00041
00042 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
00043 ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
00044 WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
00045 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
00046 ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
00047 (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
00048 LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
00049 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00050 (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
00051 SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
00052 */
00053
00054 #if __cplusplus == 201103L
00055
00056 #include <iostream>
00057 #include <iomanip>
00058 #include <cmath>
00059 #include <cstring>
00060
00061 #include "mex.h"
00062
00063 #include "mtk.h"
00064
00065 using namespace std;
00066
00067 static int MTKDivlD(int order_accuracy,
00068                    mtk::Real tau,
00069                    mtk::Real *outS,
00070                    mtk::Real *outM,
00071                    mtk::Real *outQ);
00072
00073 static int MTKDivlDReturnAsDenseMatrix(int order_accuracy,
00074                                       mtk::Real west_bndy_x,
00075                                       mtk::Real east_bndy_x,
00076                                       int num_cells,
00077                                       mtk::Real tau,
00078                                       mtk::Real *outMatrix);
00079
00080 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[]) {
00081
00082     int order_accuracy; // kk inside MTK.
00083     int num_cells;      // scaler to matrix size.
00084
00085     mtk::Real *outMatrix; // Where to place our output.
00086     mtk::Real *outS;      // output Stencil.
00087     mtk::Real *outQ;      // output Weights.
00088     mtk::Real *outM;      // output Mimetic Coefficients.
00089
00090     mtk::Real west_bndy_x; // Start of the Interval.
00091     mtk::Real east_bndy_x; // End of the Interval.
00092     mtk::Real tau;         // Mimetic tolerance (optional).
00093
00094     // if (nrhs >= 1 && nrhs <= 2) {
00095     //
00096     //     order_accuracy = mxGetScalar(prhs[0]);
00097     //
00098     //     if (nrhs == 1 && order_accuracy >= 8) {
00099     //         std::cout << "Default mimetic threshold set as 1e-6."<<std::endl;
00100     //         tau = 1e-6;
00101     //     }

```



```

00102 //      if (nrhs == 2 && order_accuracy < 8) {
00103 //          std::cout << "Order of accuracy does not need mimetic threshold. If provided it will be
00104 //      ignored."<<std::endl;
00105 //          tau = mxGetScalar(prhs[1]);
00106 //      }
00107 //      if (nrhs == 2 && order_accuracy >= 8) {
00108 //          tau = mxGetScalar(prhs[1]);
00109 //      }
00110 //      if (nlhs != 3) {
00111 //          mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs","3 outputs are needed: S, M, Q.");
00112 //      }
00113 //      if (nlhs == 3) {
00114 //          // Stencil
00115 //          plhs[0] = mxCreateDoubleMatrix((mwSize) 1,(mwSize) order_accuracy,mxREAL);
00116 //          //Mimetic coefficients Rows
00117 //          plhs[1] = mxCreateDoubleMatrix( (mwSize) (order_accuracy/2-1), (mwSize) (3*order_accuracy/ 2),
00118 //              mxREAL);
00119 //          // Weights
00120 //          plhs[2] = mxCreateDoubleMatrix((mwSize) 1,(mwSize) order_accuracy, mxREAL);
00121 //          outS = mxGetPr(plhs[0]);
00122 //          outM =mxGetPr(plhs[1]);
00123 //          outQ = mxGetPr(plhs[2]);
00124 //          mtk_ld_div(order_accuracy, tau, outS ,outM,outQ);
00125 //      }
00126 //      } else {
00127 //          if (nrhs > 5) {
00128 //              mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs", "At most 5 inputs are permitted: (order, a, b,
00129 //              cells, tau).");
00130 //          }
00131 //          if (nrhs < 4) {
00132 //              mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs", "At least 4 inputs are required: (order, a, b,
00133 //              cells).");
00134 //          }
00135 //          if (nlhs != 1) {
00136 //              mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs","One output is needed.");
00137 //          }
00138 //          order_accuracy = mxGetScalar(prhs[0]);
00139 //          west_bndy_x = mxGetScalar(prhs[1]);
00140 //          east_bndy_x = mxGetScalar(prhs[2]);
00141 //          num_cells = mxGetScalar(prhs[3]);
00142 //          if (num_cells < 3*order_accuracy - 1) {
00143 //              mexErrMsgIdAndTxt("MyToolbox:arrayProduct:nlhs", "Number of cells too small for required order.");
00144 //          }
00145 //          if (order_accuracy >= 8) {
00146 //              std::cout << "Order of accuracy too high. Mimetic threshold will be applied." << std::endl;
00147 //          }
00148 //          if (nrhs == 5 && order_accuracy >= 8) {
00149 //              tau = mxGetScalar(prhs[4]);
00150 //          }
00151 //          if (nrhs == 5 && order_accuracy < 8 ) {
00152 //              std::cout << "Order of accuracy does not need mimetic threshold. If provided it will be
00153 //              ignored."<<std::endl;
00154 //              tau = mxGetScalar(prhs[4]);
00155 //          }
00156 //          if (nrhs == 4 && order_accuracy >= 8) {
00157 //              //Default threshold, used only if order >=8
00158 //              std::cout << "Default mimetic threshold set as 1e-6."<<std::endl;
00159 //              tau = 1.0e-6;
00160 //          }
00161 //          plhs[0] = mxCreateDoubleMatrix((mwSize) (num_cells + 2),
00162 //              (mwSize) (num_cells + 1),
00163 //              mxREAL);
00164 //      }

```

```

00178 //      outMatrix =  mxGetPr(plhs[0]);
00179 //
00180 //      MTKDiv1DReturnAsDenseMatrix(order_accuracy, west_bndy_x, east_bndy_x, num_cells, tau, outMatrix);
00181 //  }
00182 }
00183
00184 static int MTKDiv1D(int order_accuracy,
00185                    mtk::Real tau,
00186                    mtk::Real *outS,
00187                    mtk::Real *outM,
00188                    mtk::Real *outQ) {
00189
00190 //      mtk::MTK_1DDiv *div = new  mtk::MTK_1DDiv(order_accuracy,tau);
00191 //
00192 //      int Number_of_Extra_Rows=order_accuracy/2-1;
00193 //
00194 //      div = div->Construct1DDiv();
00195 //
00196 //      if (div == nullptr) {
00197 //          return EXIT_FAILURE;
00198 //      }
00199 //
00200 //      mtk::Real *ww;
00201 //      mtk::Real *ss;
00202 //      mtk::MTK_DenseMatrix *ee;
00203 //      mtk::MTK_DenseMatrix *eeT;
00204 //
00205 //      ss = div->ReturnStencil();
00206 //
00207 //      ww = div->ReturnWeights();
00208 //
00209 //      for(auto ii = 0; ii < order_accuracy; ii++) {
00210 //          outQ[ii]= ww[ii];
00211 //          outS[ii]= ss[ii];
00212 //      }
00213 //
00214 //      ee = div->ReturnMimeticCoefficients();
00215 //      eeT= new mtk::MTK_DenseMatrix(3*order_accuracy/2,Number_of_Extra_Rows);
00216 //      eeT->Transpose(*ee,*eeT);
00217 //
00218 //      for(auto ii=0;ii<3*order_accuracy/2*Number_of_Extra_Rows;ii++) {
00219 //          out M[ii]=eeT->DenseMatrix_To_Real(*eeT) [ii];
00220 //      }
00221 //
00222 //      delete [] ww;
00223 //      delete [] ss;
00224 //      delete [] ee;
00225 }
00226
00227 static int MTKDiv1DReturnAsDenseMatrix(int order_accuracy,
00228                                       mtk::Real west_bndy_x,
00229                                       mtk::Real east_bndy_x,
00230                                       int num_cells,
00231                                       mtk::Real tau,
00232                                       mtk::Real *outMatrix) {
00233
00234 //      int nn = num_cells + 2;
00235 //      int mm = num_cells + 1;
00236 //
00237 //      mtk::MTK_DenseMatrix *dd;
00238 //      mtk::MTK_DenseMatrix *ddT;
00239 //
00240 //      mtk::MTK_1DDiv *div = new  mtk::MTK_1DDiv(order_accuracy,tau);
00241 //
00242 //      div = div->Construct1DDiv();
00243 //
00244 //      ddT = new mtk::MTK_DenseMatrix(mm,nn);
00245 //
00246 //      if (div == nullptr) {
00247 //          std::cout << "Problem constructing the matrix... " << std::endl;
00248 //          return EXIT_FAILURE;
00249 //      }
00250 //
00251 //      dd = div->ReturnAsMatrix(num_cells, west_bndy_x, east_bndy_x);
00252 //
00253 //      dd->Transpose(*dd,*ddT);
00254 //
00255 //      for(auto ii = 0; ii < mm*nn; ii++) {
00256 //          outMatrix[ii]=ddT->DenseMatrix_To_Real(*ddT) [ii];
00257 //      }
00258 //

```

```

00259 //   delete [] ddT;
00260 }
00261
00262 #endif

```

## 9.9 mexsrc/MMTKDiv1D.m File Reference

### 9.10 MMTKDiv1D.m

```

00001 % MMTKDiv1D Create a mimetic one-dimensional divergence operator.
00002 %
00003 %   div = MMTKDiv1D(k, w, e, n, tau) Returns the divergence order k as a n + 2
00004 %                                   by n + 1 matrix, in the interval (w,e) with
00005 %                                   discretized using n cells with the given
00006 %                                   mimetic threshold tau (optional).
00007 %
00008 %   [s, m, q] = MMTKDiv1D(k, tau) Returns the divergence order k as the
00009 %                                   interior stencil s, mimetic coefficients m,
00010 %                                   mimetic weights q and mimetic threshold tau
00011 %                                   (optional).
00012 %
00013 % For more information, see
00014 % <a href="http://www.csrc.sdsu.edu/mimetic-book/">
00015 % Mimetic Discretization Methods</a>.
00016 %

```

### 9.11 README.md File Reference

### 9.12 README.md

```

00001 # README File for the Mimetic Methods Toolkit (MTK)
00002
00003 By: **Eduardo J. Sanchez, Ph.D. - esanchez at mail dot sdsu dot edu**
00004
00005
00006 ## 1. Description
00007
00008 We define numerical methods that are based on discretizations preserving the
00009 properties of their continuum counterparts to be **mimetic**.
00010
00011 The **Mimetic Methods Toolkit (MTK)** is a C++ library for mimetic numerical
00012 methods. It is arranged as a set of classes for **mimetic quadratures**,
00013 **mimetic interpolation**, and **mimetic discretization** methods for the
00014 numerical solution of ordinary and partial differential equations.
00015
00016 This collection of **MATLAB Wrappers for the MTK (MMTK)** allows developers to
00017 invoke the MTK from a MATLAB environment.
00018
00019
00020
00021 ## 2. Dependencies
00022
00023 This README assumes all of these dependencies are installed in the following
00024 folder:
00025
00026 ```
00027 $(HOME)/Libraries/
00028 ```
00029
00030 In this version, the MTK optionally uses ATLAS-optimized BLAS and LAPACK
00031 routines for the internal computation on some of the layers. However, ATLAS
00032 requires both BLAS and LAPACK in order to create their optimized distributions.
00033 Therefore, the following dependencies tree arises:
00034
00035 ### For Linux and OS X:
00036
00037 1. MATLAB R2014a or greater - Available from: http://www.mathworks.com/
00038

```

```

00039 1. MTK - Available from: http://csrc.sdsu.edu/mtk/
00040
00041
00042 ## 3. Installation
00043
00044 The following instructions assume MATLAB R2014a or greater.
00045
00046 You have two options, either follow the instructions given, **expert install**,
00047 or use the provided **patch file**, **naive install**.
00048
00049 ### EXPERT PART 1. CONFIGURATION OF THE MEX COMPILER.
00050
00051 From your MATLAB prompt, type:
00052
00053 ```
00054 >> mex -setup C++
00055 MEX configured to use 'g++' for C++ language compilation.
00056 Warning: The MATLAB C and Fortran API has changed to support MATLAB
00057 variables with more than 2^32-1 elements. In the near future
00058 you will be required to update your code to utilize the
00059 new API. You can find more information about this at:
00060
00061 http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-
00062 64-bit-api.html.
00063 >>
00064 ```
00065
00066 ### EXPERT PART 2. CONFIGURATION OF THE MEX COMPILER SETUP FILE.
00067
00068 The previous step creates the following file:
00069
00070 ```
00071 $(HOME)/.matlab/R2014a/mex_C++_glnxa64.xml
00072 ```
00073
00074 The purpose of this section is to configure the MATLAB R2014a Mex compiler so
00075 that it can work with the latest C++ standard (C++11).
00076
00077 Please execute the following changes on the aforementioned file (line numbers
00078 may differ):
00079
00080 Lines 26 and 27:
00081
00082 ```
00083 CMDLINE1="$CXX -std=c++11 -c $DEFINES $INCLUDE $CXXFLAGS $OPTIM $SRC -o $OBJ"
00084 CMDLINE2="$LDXX -std=c++11 $LDFLAGS $LDTYPE $LINKOPTIM $LINKEXPORT $OBJS $CXXLIBS $LINKLIBS -o $EXE"
00085 ```
00086
00087 Line 33:
00088
00089 ```
00090 CXXFLAGS="-std=c++11 -ansi -fexceptions -fPIC -fno-omit-frame-pointer -pthread"
00091 ```
00092
00093 Line 35 and 26:
00094
00095 ```
00096 CXXOPTIMFLAGS="-std=c++11 -O -DNDEBUG"
00097 CXXDEBUGFLAGS="-std=c++11 -g"
00098 ```
00099
00100 Line 38:
00101
00102 ```
00103 LDXX="gfortran"
00104 ```
00105
00106 Line 59:
00107
00108 ```
00109 CXXFLAGS="-std=c++11 -ansi -pthread"
00110 ```
00111
00112 ### NAIVE PART 1. CONFIGURATION OF THE MEX COMPILER.
00113
00114 From your MATLAB prompt, type:
00115
00116 ```
00117 >> mex -setup C++
00118 MEX configured to use 'g++' for C++ language compilation.
00119 Warning: The MATLAB C and Fortran API has changed to support MATLAB

```

```

00120     variables with more than 2^32-1 elements. In the near future
00121     you will be required to update your code to utilize the
00122     new API. You can find more information about this at:
00123
00124     http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-
00125     64-bit-api.html.
00126     >>
00127     ```
00128
00129     ### NAIVE PART 2. CONFIGURATION OF THE MEX COMPILER SETUP FILE.
00130
00131     ```
00132     cd $HOME/.matlab/R2014a
00133     chmod +w mex_C++_glnxa64.xml
00134     patch < mex_C++_glnxa64.patch
00135     chmod -w mex_C++_glnxa64.xml
00136     ```
00137     Exit terminal, and restart MATLAB. You can use C++11 to create MEX files now!
00138
00139     ### PART 3: CONFIGURATION OF THE MAKEFILE.
00140
00141     The following steps are required the build and test the MTK. Please use the
00142     accompanying 'Makefile.inc' file, which should provide a solid template to
00143     start with. The following command provides help on the options for make:
00144
00145     ```
00146     $ make help
00147     -----
00148     Makefile for the MMTK.
00149
00150     Options are:
00151     - all: builds the library, the tests, and examples.
00152
00153     - gendoc: generates the documentation for the library.
00154
00155     - clean: cleans ALL the generated files.
00156     -----
00157     ```
00158
00159     ### PART 4. BUILD THE MMTK.
00160
00161     From your shell, at the base folder of the MMTK, just type:
00162
00163     ```
00164     make
00165     ```
00166
00167     If successful you'll read:
00168
00169     ```
00170     ----- Library created! Check in /home/ejspeiro/Dropbox/MTK/lib
00171     ```
00172
00173     -----
00174     ## 4. Frequently Asked Questions
00175
00176     Q: Why haven't you guys implemented GBS to build the library?
00177     A: I'm on it as we speak! ;)
00178
00179     Q: When will the other flavors be ready?
00180     A: Soon! I'm working on getting help on developing those.
00181
00182     Q: Is there any main reference when it comes to the theory on Mimetic Methods?
00183     A: Yes! Check: http://www.csrc.sdsu.edu/mimetic-book
00184
00185     Q: Do I need to generate the documentation myself?
00186     A: You can if you want to... but if you DO NOT want to, just go to our website.
00187
00188     -----
00189     ## 5. Contact, Support, and Credits
00190
00191     The MTK is developed by researchers and adjuncts to the
00192     [Computational Science Research Center (CSRC)](http://www.csrc.sdsu.edu/)
00193     at [San Diego State University (SDSU)](http://www.sdsu.edu/).
00194
00195     Developers are members of:
00196
00197     1. Mimetic Numerical Methods Research and Development Group.
00198     2. Computational Geoscience Research and Development Group.
00199     3. Ocean Modeling Research and Development Group.
00200

```

```
00201 Currently the developers are:
00202
00203 - **Eduardo J. Sanchez, Ph.D. - esanchez at mail dot sdsu dot edu** - @ejspeiro
00204 - Jose E. Castillo, Ph.D. - jcastillo at mail dot sdsu dot edu
00205 - Guillermo F. Miranda, Ph.D. - unigrav at hotmail dot com
00206 - Christopher P. Paolini, Ph.D. - paolini at engineering dot sdsu dot edu
00207 - Angel Boada.
00208 - Johnny Corbino.
00209 - Raul Vargas-Navarro.
00210
00211 Finally, please feel free to contact me with suggestions or corrections:
00212
00213 **Eduardo J. Sanchez, Ph.D. - esanchez at mail dot sdsu dot edu** - @ejspeiro
00214
00215 Thanks and happy coding!
```

# Index

all

Div1D.m, [23](#)

clc

Div1D.m, [23](#)

Div1D.m

all, [23](#)

clc, [23](#)

shortG, [23](#)

examples/Div1D.m, [23](#)

include/mmtk.h, [24](#)

Makefile.inc, [25](#)

mexsrc/MMTKDiv1D.cc, [27](#)

mexsrc/MMTKDiv1D.m, [31](#)

README.md, [31](#)

shortG

Div1D.m, [23](#)