

## Nektar++: Library design

Chris Cantwell, David Moxey, Mike Kirby, Spencer Sherwin

Nektar++ Workshop  
Imperial College London  
7th July 2015

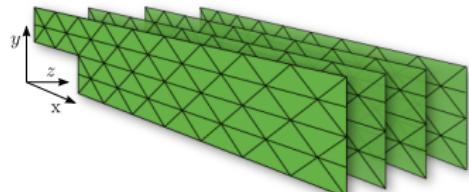
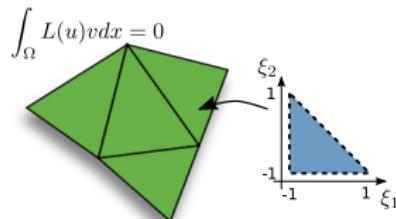
# Overview

- Library design overview
- Collections
- Developer Practice
- User Guide and Developer Guide

# Nektar++: Objectives

## Mathematical Construction

Expose different discretisations (CG, DG) by combining and reusing low-level elemental mathematical constructs.

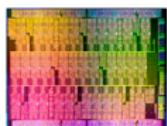


Cross-dimensional support: 1D, 2D and 3D

Retain and exploit domain symmetries and embeddings (homogeneous, cylindrical, manifold)

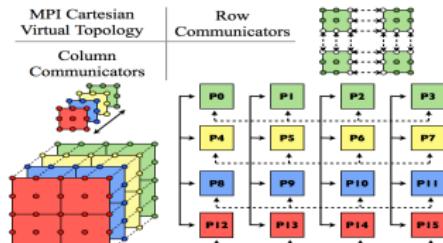
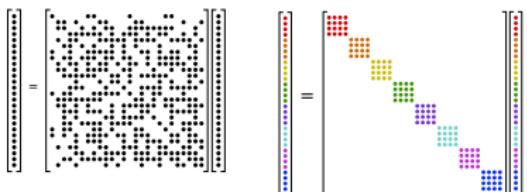
## Computational Implementation

Challenge high-/low-order boundaries while maintaining efficiency.

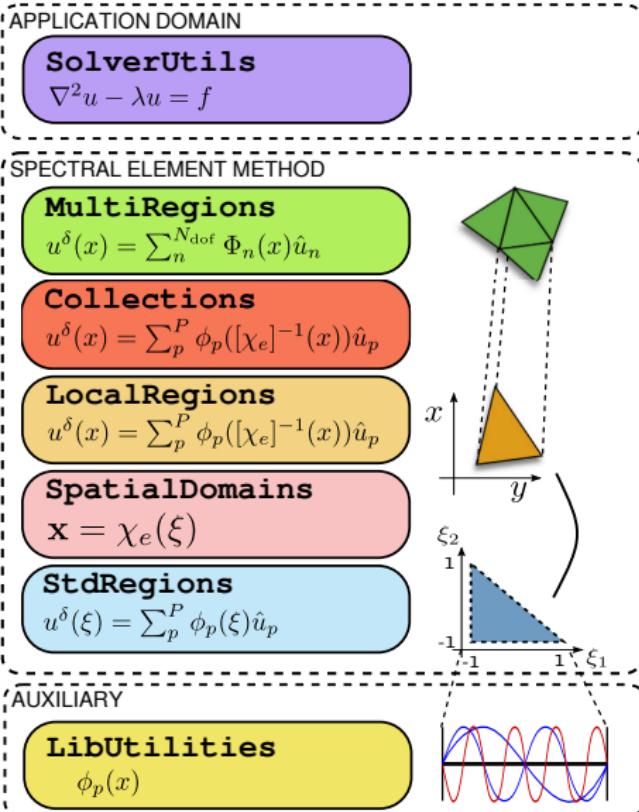


Bridge current and future hardware diversity.

Achieve flexible HPC scalability and performance through hybrid parallelism.

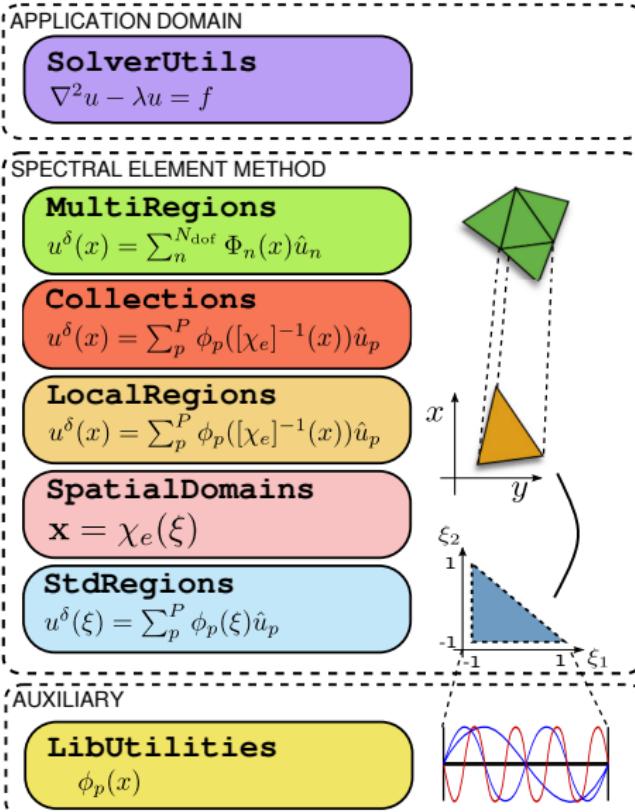


# Nektar++: Design



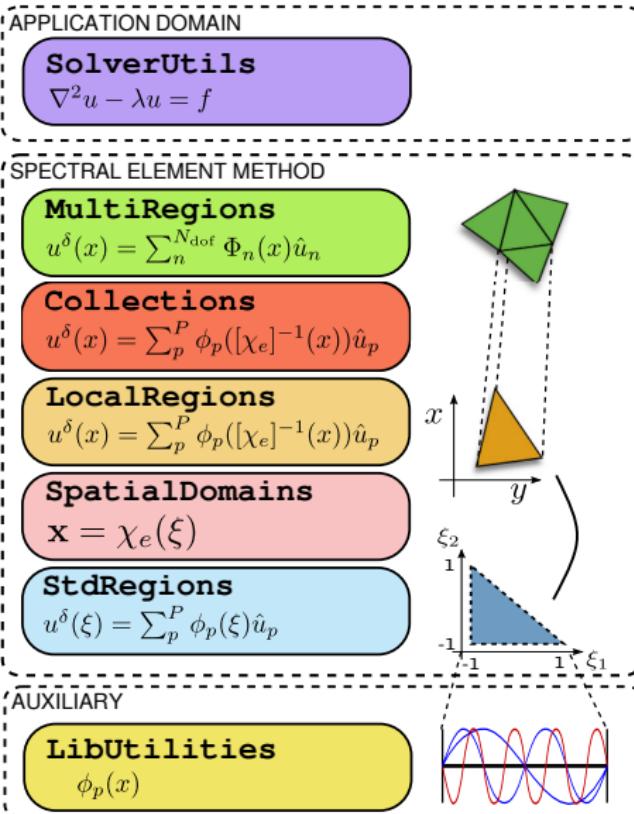
- Stack of libraries
- Build up the spectral/ $hp$  element method systematically

# Nektar++: Design



- Stack of libraries
- Build up the spectral/ $hp$  element method systematically
- Core mathematical abstractions at bottom
- Complete physical domain description and operators at top

# Nektar++: Design



- Stack of libraries
- Build up the spectral/*hp* element method systematically
- Core mathematical abstractions at bottom
- Complete physical domain description and operators at top
- Encapsulate complexities of the spectral/*hp* element formulation at the top levels to improve accessibility of methods
- Allow users to still engage at lower level depending on their requirements (e.g. for methods research)

# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$

# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$



Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = \boxed{(f, v)}$

# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

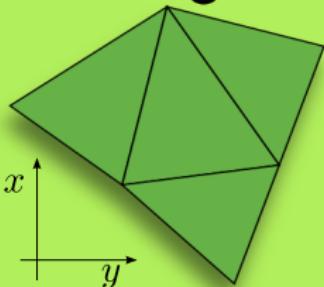
Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$



Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

## MultiRegions



$$\mathbf{f}[i] = \int_{\Omega} \Phi_i(x) f(x) dx$$

# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$

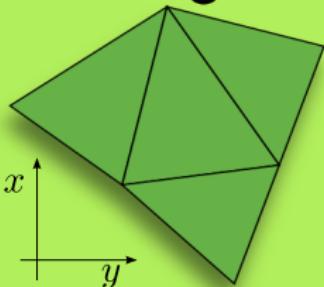


Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

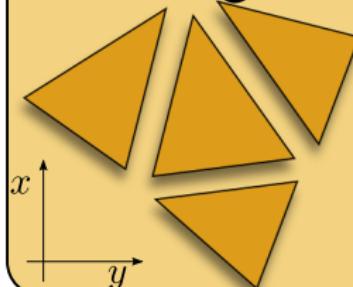
$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

## MultiRegions



$$\mathbf{f}[i] = \int_{\Omega} \Phi_i(x) f(x) dx = \sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x) dx$$

## LocalRegions



# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$

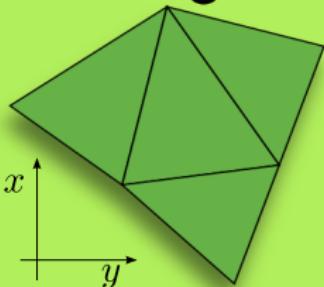


Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

## MultiRegions

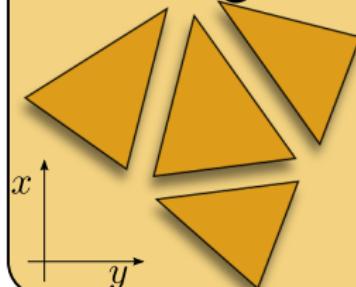


$$\mathbf{f}[i] = \int_{\Omega} \Phi_i(x) f(x) dx$$

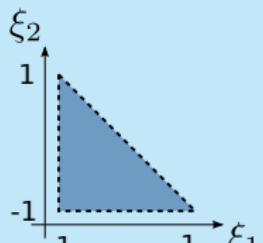
$$= \sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x) dx$$

$$= \sum_e \sum_p \int_{\Omega^e} \phi_p(\chi^e(\xi)) f(\chi^e(\xi)) J^e d\xi$$

## LocalRegions



## StdRegions



# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

$$\nabla^2 u + \lambda u = f$$

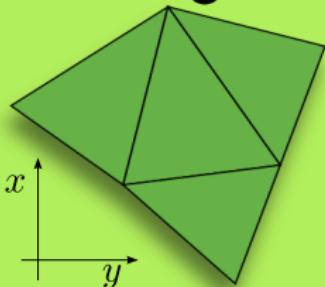


Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

## MultiRegions

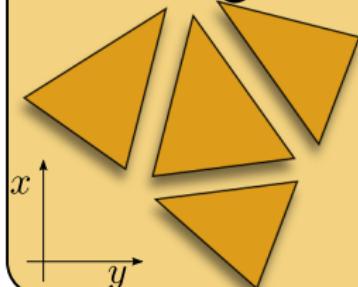


$$\mathbf{f}[i] = \int_{\Omega} \Phi_i(x) f(x) dx$$

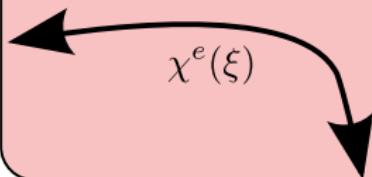
$$= \sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x) dx$$

$$= \sum_e \sum_p \int_{\Omega^e} \phi_p(\chi^e(\xi)) f(\chi^e(\xi)) J^e d\xi$$

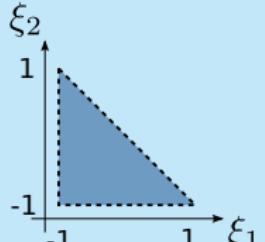
## LocalRegions



## SpatialDomains



## StdRegions



# Nektar++: Operator construction

Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem:

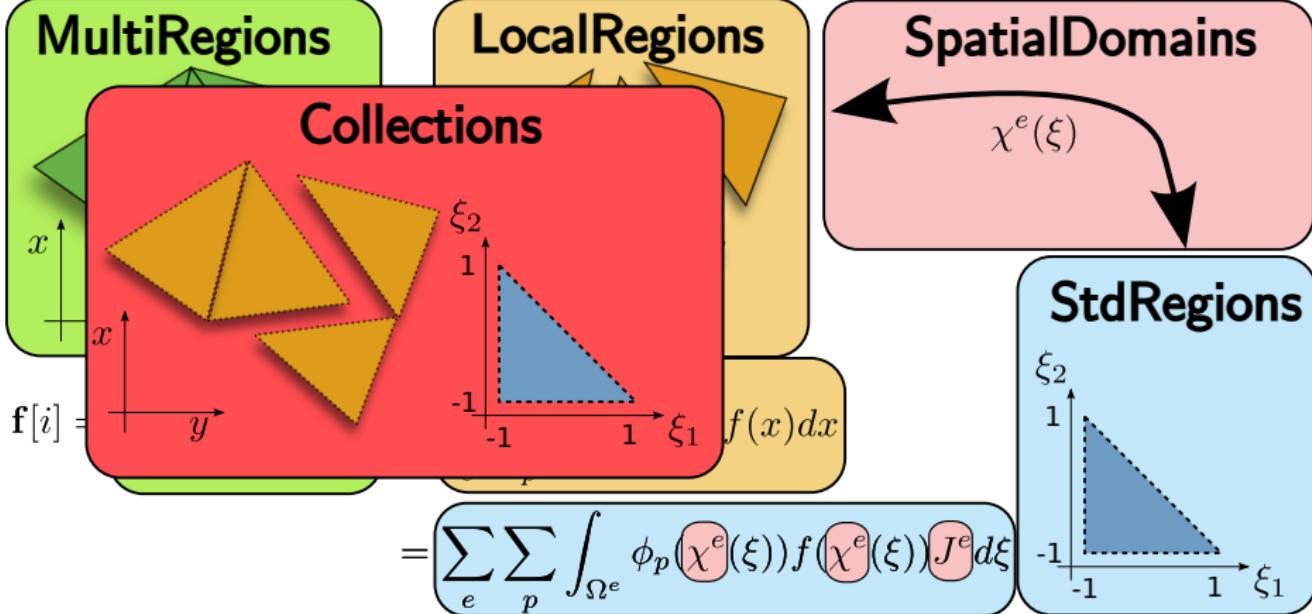
$$\nabla^2 u + \lambda u = f$$



Weak form + IBP:  $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

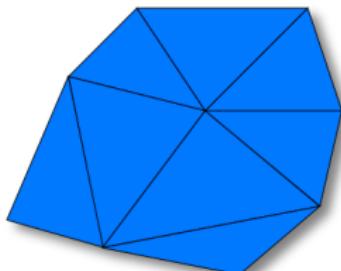
$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$



# Implementation strategies



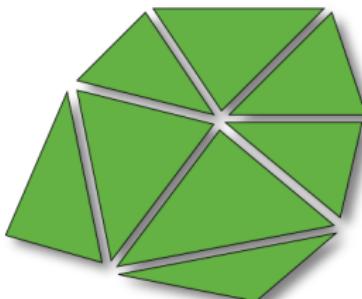
# Implementation strategies



Global Strategy



$$\begin{array}{c|c|c} \vdots & = & \vdots \\ & \text{A sparse matrix with many zero entries and scattered non-zero entries.} & \\ \vdots & = & \vdots \end{array}$$

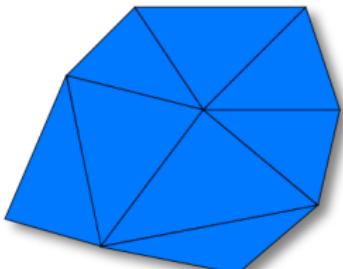


Local Strategy



$$\begin{array}{c|c|c} \vdots & = & \vdots \\ & \text{A sparse matrix with a banded structure where non-zero entries are colored in a diagonal pattern (red, orange, yellow, green, blue, purple).} & \\ \vdots & = & \vdots \end{array}$$

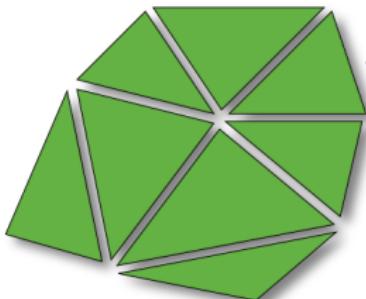
# Implementation strategies



Global Strategy

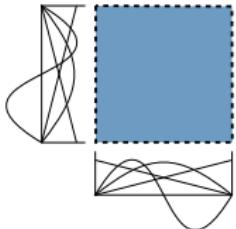
$\mathcal{A}$

$$= \begin{array}{|c|} \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \end{array}$$



Local Strategy

$$= \begin{array}{|c|} \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot & \cdot \\ \hline \end{array}$$



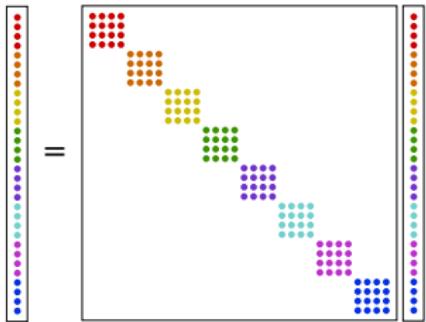
Sum-factorisation

$$= \begin{array}{|c|} \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \end{array} \quad \begin{array}{|c|} \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \end{array} \quad \begin{array}{|c|} \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \cdot & \cdot & \cdot \\ \hline \end{array}$$

Localised Memory Access  
Increasing Polynomial Order

# Collections: Design

## Local Matrix



## StdMat

1. Apply Jacobian (L1)

$$\begin{array}{c|c} \text{Color Vector} & = \\ \hline \text{Red, Orange, Yellow, Green, Blue, Purple} & \odot \end{array}$$

A diagram showing the multiplication of a color vector by a scalar. On the left is a vertical vector of colored dots (red, orange, yellow, green, blue, purple). An equals sign follows it. To the right is a small circle with a dot inside, representing scalar multiplication. Further to the right is another vertical vector of colored dots, identical to the first.

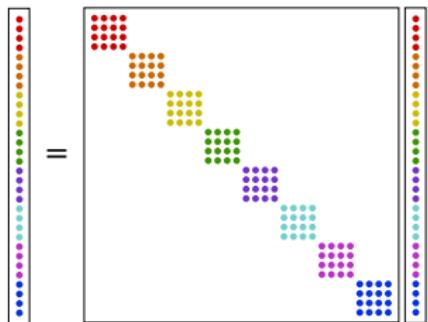
2. L3 Multiply by ref. matrix

$$\begin{array}{c|c} \text{Color Vector} & = \\ \hline \text{Red, Orange, Yellow, Green, Blue, Purple} & \times \end{array}$$

A diagram showing the multiplication of a color vector by a reference matrix. On the left is a vertical vector of colored dots (red, orange, yellow, green, blue, purple). An equals sign follows it. To the right is a small circle with a dot inside, representing scalar multiplication. Further to the right is a 3x3 grid of black dots, representing a reference matrix. To the right of the grid is another vertical vector of colored dots, identical to the first.

# Collections: Design

## Local Matrix



## StdMat

### 1. Apply Jacobian (L1)

The diagram illustrates the first step of the StdMat design. On the left is a vertical vector with colored dots. An equals sign follows it. To the right is a square matrix with colored circles. A dot product symbol ( $\odot$ ) is positioned between the vector and the matrix. To the right of the dot product is another vertical vector with colored dots.

### 2. L3 Multiply by ref. matrix

The diagram illustrates the second step of the StdMat design. On the left is a vertical vector with colored dots. An equals sign follows it. To the right is a square matrix with black dots. To its right is another vertical vector with colored dots.

## IterPerExp

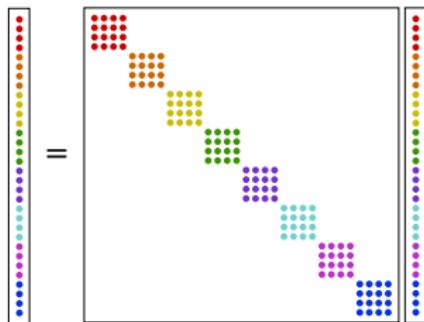
1. Apply Jacobian (L1)
2. L2 multiply by  
ref. matrix

for i = 1:N

The diagram shows a vertical vector with three red dots on the left, followed by an equals sign, then a square matrix with a 3x3 grid of black dots, and finally a vertical vector with three red dots on the right.

# Collections: Design

## Local Matrix



## StdMat

### 1. Apply Jacobian (L1)

$$\begin{array}{c|c} \text{Colorful Vector} & = \\ \hline \text{Colorful Vector} & \odot \end{array}$$

### 2. L3 Multiply by ref. matrix

$$\begin{array}{c|c} \text{Colorful Vector} & = \\ \hline \text{Black Matrix} & \text{Colorful Vector} \end{array}$$

## IterPerExp

1. Apply Jacobian (L1)
2. L2 multiply by ref. matrix

for i = 1:N

$$\begin{array}{c|c} \text{Vertical Vector} & = \\ \hline \text{Matrix} & \text{Vertical Vector} \end{array}$$

## SumFac (Quad)

1. Apply Jacobian (L1)
2. L2 multiply for first dim ref. matrix

$$\begin{array}{c|c} \text{Banded Matrix} & = \\ \hline \text{Banded Matrix} & \text{Banded Matrix} \end{array}$$

3. L2 multiply for second dim ref. matrix
- for i = 1:N

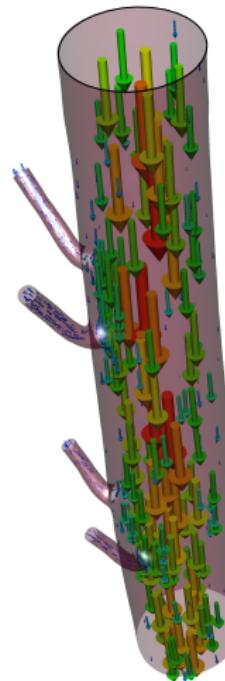
$$\begin{array}{c|c} \text{Banded Matrix} & = \\ \hline \text{Banded Matrix} & \text{Banded Matrix} \end{array}$$

# Test Case

Intercostal Pair

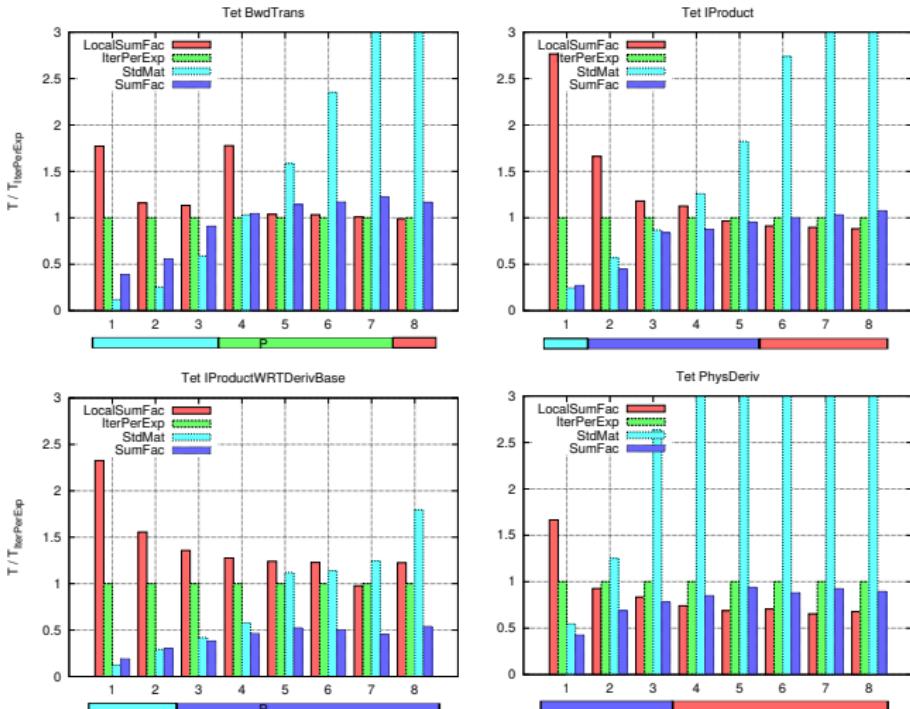
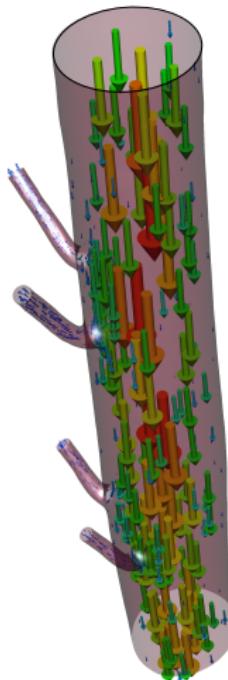
41k tets

21k prisms



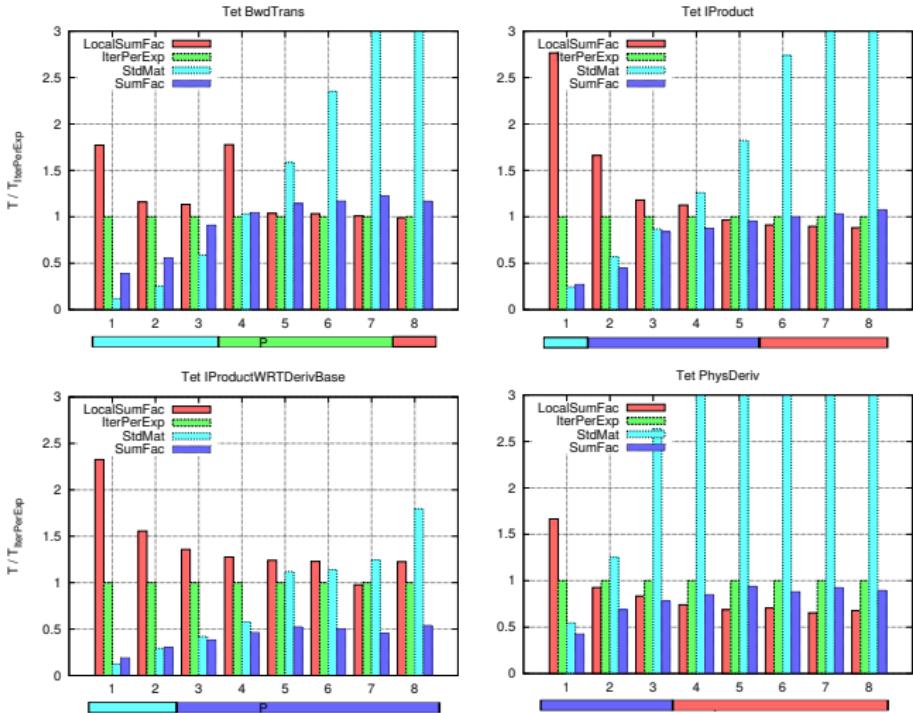
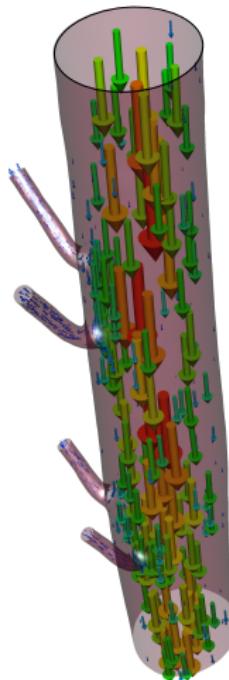
# Test Case

Intercostal Pair  
41k tets  
21k prisms



# Test Case

Intercostal Pair  
41k tets  
21k prisms



StdMat most effective at low order

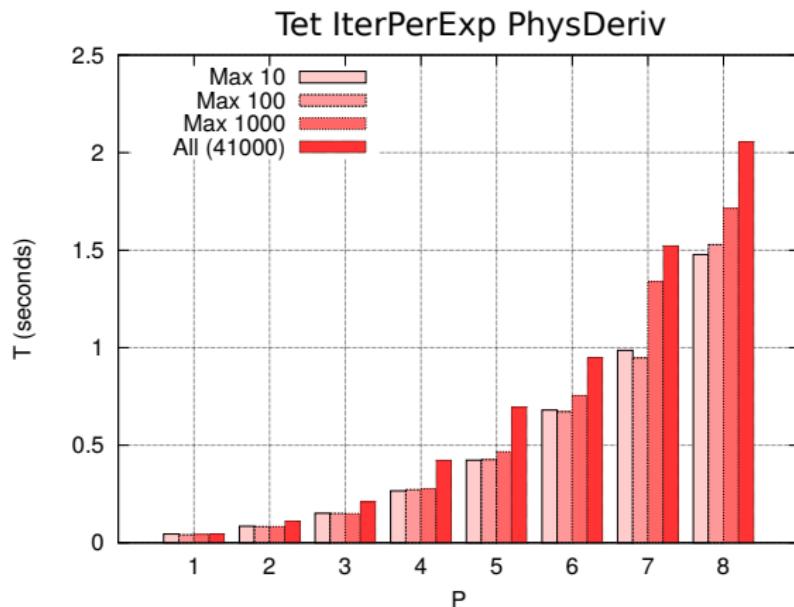
Collections less effective at high order for most operators

PhysDeriv benefits from SumFac even at low P

Similar trends observed for Prisms

# Collection Size Dependence

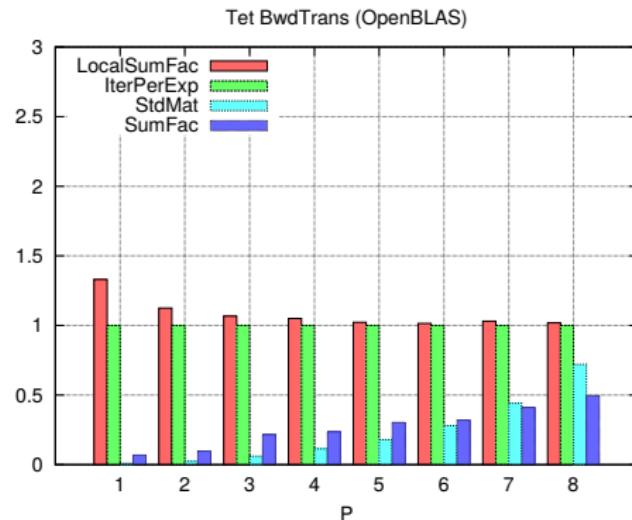
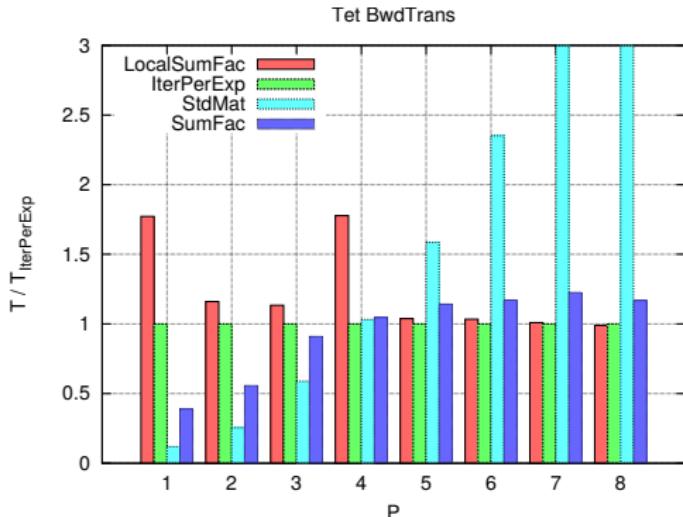
Limit size of collections to 10, 100 or 1000 elements



Negligible dependence on collection size for < 1000 elements

Reduced performance for large numbers of elements

# BLAS Implementation



OpenBLAS performs best for large matrix operations

Significant gains for StdMat and SumFac at high P

Importance of auto-tuning

## Developer Practice

What development practices support a large multi-platform collaborative software project such as Nektar++?

# Developer Practice

What development practices support a large multi-platform collaborative software project such as Nektar++?

- Version-control (Git + GitLab)

<http://gitlab.nektar.info>

```
git clone https://gitlab.nektar.info/nektar/nektar.git
```



# Developer Practice

Nektar / Nektar

Search in this project

Project Files Commits Network Graphs Merge Requests 6 Snippets Settings

Nektar++ - Edit

★ Star 0 Fork 5

SSH HTTPS git@gitlab.nektar.info:nekta/nektar.git public Download zip

Activity Readme 7,639 commits 52 branches 7 tags 264.27 MB

Chris Cantwell commented on merge request #475 at Nektar / Nektar 15 minutes ago  
@ym1008, @y.bao, @d.serson, @dmoxey: Could I remind you to try and complete this review by the end of today so we can get this merged?

Spencer Sherwin pushed to branch fix/LowEnergyTransformOpt at Nektar / Nektar a day ago  
b3cfBeab First working version of optimised version of Low energy Transform...

Spencer Sherwin pushed to branch fix/LowEnergyTransformOpt at Nektar / Nektar 3 days ago  
4a54fa45 Merge branch 'master' into fix/LowEnergyTransformOpt  
6c3be247 Merge branch 'feature/threading' into 'master'  
... and 70 more commits. Compare → 83904242...4a54fa45

Spencer Sherwin pushed to branch fix/LowEnergyTransformOpt at Nektar / Nektar 3 days ago  
83904242 Merge branch 'master' into fix/LowEnergyTransformOpt  
69f2992c Merge branch 'fix/valgrind-collections-tests' into 'master'  
... and 988 more commits. Compare → ce561519...83904242

Spencer Sherwin pushed to branch feature/FieldConvertCleanUp at Nektar / Nektar 3 days ago  
fb2e59a3 Misused off on previous commit

Spencer Sherwin pushed to branch feature/FieldConvertCleanUp at Nektar / Nektar 3 days ago  
ad41790d Replaced print with an ASSERTL0

Spencer Sherwin commented on merge request #475 at Nektar / Nektar 4 days ago  
Done

Compare code Version: 4.0.1

Created on Sep 26, 2012  
Owned by Nektar group

# Developer Practice

Nektar / Nektar

Search in this project

Project Files Commits Network Graphs Merge Requests 6 Snippets Settings

master nektar / + Download zip History

Name	Last Update	Last Commit >	History
cmake	11 days ago	Spencer Sherwin Merge branch 'master' into feature/MeshConvertFullStarInput	
docs	6 days ago	Chris Cantwell Merge branch 'feature/MeshConvertFullStarInput' into 'master'	
library	4 days ago	Chris Cantwell Merge branch 'feature/threading' into 'master'	
pkg	5 months ago	Chris Cantwell Modified targets file writing for earlier versions of CMake.	
solvers	7 days ago	David Moxey Merge branch 'feature/cell-history-points' into 'master'	
templates	about a year ago	Kilian Lackhove LibUtilities must be linked against rt for non-windows-systems	
tests	8 months ago	David Moxey Bump TinyXML to 2.6.2	
utilities	4 days ago	Chris Cantwell Merge branch 'feature/threading' into 'master'	
.gitignore	5 months ago	Kilian Lackhove extend .gitignore	
CMakeLists.txt	about a month ago	Spencer Sherwin Working version with search for libccmio.a and libadf.a	
LICENSE	5 years ago	Chris Cantwell Tidied up PreProcessing utilities.	
README.md	5 months ago	David Moxey Move README -> README.md	
VERSION	6 months ago	Chris Cantwell Updated version to 4.0.1	

README.md

## Nektar++

Nektar++ is an open-source software framework designed to support the development of high-performance scalable solvers for partial differential equations (PDEs) using the spectral/hp element method.

This package consists of a set of libraries (the framework) and a number of pre-written PDE solvers for a selection of application domains.

The software and User Guide is available for download from <http://www.nektar.info/>.

# Developer Practice

What development practices support a large multi-platform collaborative software project such as Nektar++?

- Version-control (Git + GitLab)



# Developer Practice

What development practices support a large multi-platform as Nektar++?

assignee: Any author: Any milestone: Any sort: Newest

Feature/field convert clean up feature/FieldConvertCl... > master

#475 authored by Chris Cantwell 56 0 v4.1.0

Feature/bessel

#472 authored by Michael Barbour

Feature/moving bodies

#448 authored by Yan Bao 20

Feature/compressible bl

#439 authored by Gianmarco Mengaldo 14

Add adiabatic wall boundary conditions to CFS

#430 authored by David Moxey 5 0 v4.1.0

Reynolds stress filter

#405 authored by David Moxey 7 0 v4.1.0

Merge Request #405  
To merge requests From feature/filterReynoldsStress into master

Open nektar/nektar#405 Created by David Moxey 7 months ago

Download as ▾ Close ⌂ Edit 0 up 0 down

Reynolds stress filter

This MR adds a Reynolds stress filter.

Assignee: David Moxey Milestone: v4.1.0

You can accept this request automatically.

Accept Merge Request Remove source-branch ⌂ modify merge commit message

If you still want to merge this request manually - use command line

Commits (5)

14048bd Merge branch 'master' into feature/filterReynoldsStress → Browse Code 5 months ago

4c950a7 Merge branch 'master' into feature/filterReynoldsStress → Browse Code 7 months ago

b832f57 Change to make Reynolds stresses inherit from average field filter so that only ... → Browse Code about a year ago

2e6a3a25 Merge branch 'master' into feature/filterReynoldsStress → Browse Code about a year ago

b9b346e Added first compiling version of Reynolds Stresses data in filter → Browse Code about a year ago

4 participants David Moxey Michael Barbour Yan Bao Spencer Sherwin

Discussion Changes

David Moxey @dmoxey · 7 months ago

Added 574 new commits:

- af3f80881 - Revert "APE: Adjusted test errors and tolerances for 3D cases"
- 3b24f1b8e - Merge branch 'fix/APE-tests' into 'master'
- 74ebef417 - Switched to using class Timer to do timing
- f3fb04ab8 - Switched timing scripts to Timer class use
- e0f2899fe - Extract surface for 3D CFS solver
- 9a0b3de7 - Move utilities to compressible flow solver, fix lots of indentation and add license
- 27419ca3c - Merge branch 'fix/ExtractSurface3DCFS' into 'master'

# Developer Practice

What development practices support a large multi-platform collaborative software project such as Nektar++?

- Version-control (Git + GitLab)

<http://gitlab.nektar.info>

```
git clone https://gitlab.nektar.info/nektar/nektar.git
```



- Tests & Continuous Integration (Buildbot)

<http://buildbot.nektar.info>

The image consists of two parts. On the left is a 'Build Status Grid' titled 'Grid View' showing a 10x10 grid of build results for various jobs. Most cells are green, indicating success, while a few are red or yellow. On the right is a detailed 'Builder Ubuntu 12.04 64-bit (full) Build #659' page. It includes sections for 'Build Properties', 'Forced Build Properties', 'Steps and Logfiles', 'Responsible Users', and 'Timeline'. The 'Build Properties' section lists various environment variables and paths. The 'Timeline' section shows the build's progress over time.

# Developer Practice

Home · Waterfall Grid · Grid Console · Builders · Recent Builds · Buildslaves · Changeresources · JSON API · About

## Grid View

Nestarc+	853c2a1acf77... in fsv/svv-symmetry	853c2a1acf77... in svv-symmetry	d18ded9f0934... in feature/MeshConvertFullStartInput	d80b4fd5840c... in feature/FieldConvertCleanUp	8c3be247a931... in master
CentOS 7 64-bit (default) (building)	OK		OK	OK	building
CentOS 7 64-bit (full) (building)	OK		OK	OK	building
Debian 7 64-bit (default) (ide, plus 1)	OK	failed	OK	OK	OK
Debian 7 64-bit (full) (building)	OK		OK	OK	building
Debian 7.0 64-bit (full) (ide, plus 1)	OK	failed	OK	OK	OK
Debian 7.0 64-bit (experimental) (building)	OK		OK	OK	building
Debian 10 64-bit (full) (building)	OK		OK	OK	building
Fedora 16 64-bit (full) (building)	OK		OK	OK	building
Fedora 20 64-bit (full) (building)	OK		OK	OK	building
Fedora 21 64-bit (full) (building)	OK		OK	OK	building
Fedora 21 64-bit (full) (building)	OK		OK	OK	building
OS X Snow Leopard (beta) (full) (building)	OK		OK	OK	building
OpenSUSE 11.1 64-bit (beta) (full) (building)	OK		OK	OK	building
OpenSUSE 11.1 64-bit (full) (building)	OK		OK	OK	building
OpenSUSE 12.2 64-bit (beta) (building)	OK		OK	OK	building
Ubuntu 12.04 64-bit (full) (building)	OK		OK	OK	building
Ubuntu 12.04 64-bit (full) (building)			OK	OK	building
Ubuntu 12.04 32-bit (full) (building)	OK		OK	OK	building
Ubuntu 14.04 32-bit (full) (building)	OK		OK	OK	building
Ubuntu 14.04 64-bit (beta) (building)	OK		OK	OK	building
Ubuntu 14.04 64-bit (full) (building)	OK		OK	OK	building
Windows 7 64-bit (default) (building)	OK		OK	OK	building
Windows XP 32-bit (default) (building)	OK		OK	OK	building

Buildbot (0.8.9) working for the Nestarc+ project.  
Page built: Mon 22 Jun 2015 11:10:04 (BST)



itlab.nektar.info  
ar/nektar.git

# Developer Practice

Builders (different OSs / config)

Grid View	Build Status				
Nestar-e	OK	OK	OK	OK	building
CentOS 7 64-bit (default) (building)	OK	OK	OK	OK	building
CentOS 7 64-bit (full) (building)	OK	OK	OK	OK	building
Debian 7 64-bit (default) (idle, plus 1)	OK	failed	OK	OK	building
Debian 7 64-bit (idle) (building)	OK	OK	OK	OK	building
Debian 7.0 64-bit (full) (idle, plus 1)	OK	failed	OK	OK	building
Debian 7.0 64-bit (building)	OK	OK	OK	OK	building
Debian 10 64-bit (default) (building)	OK	OK	OK	OK	building
Fedora 20 64-bit (full) (building)	OK	OK	OK	OK	building
Fedora 20 64-bit (idle) (building)	OK	OK	OK	OK	building
Fedora 21 64-bit (full) (building)	OK	OK	OK	OK	building
Fedora 21 64-bit (idle) (building)	OK	OK	OK	OK	building
Fedora 21 64-bit (idle) (building)	OK	OK	OK	OK	building
OS X Snow Leopard (idle) (building)	OK	OK	OK	OK	building
OpenSUSE 13.1 64-bit (default) (building)	OK	OK	OK	OK	building
OpenSUSE 13.1 64-bit (full) (building)	OK	OK	OK	OK	building
OpenSUSE 13.2 64-bit (building)	OK	OK	OK	OK	building
OpenSUSE 13.2 64-bit (full) (building)	OK	OK	OK	OK	building
Ubuntu 12.04 64-bit (idle) (building)	OK	OK	OK	OK	building
Ubuntu 12.04 64-bit (idle) (building)	OK	OK	OK	OK	building
Ubuntu 12.04 64-bit (idle) (building)	OK	OK	OK	OK	building
Ubuntu 12.04 32-bit (idle) (building)	OK	OK	OK	OK	building
Ubuntu 14.04 32-bit (full) (building)	OK	OK	OK	OK	building
Ubuntu 14.04 64-bit (idle) (building)	OK	OK	OK	OK	building
Windows 7 64-bit (default) (building)	OK	OK	OK	OK	building
Windows XP 32-bit (default) (building)	OK	OK	OK	OK	building

Builds (latest on right)



# Developer Practice

## Builder Ubuntu 12.04 64-bit (full) Build #659

### Results:

Failed tests

### SourceStamp:

Branch feature/threading  
Got Revision 3176edb8fc5bc697a832c660ff0d64662963566d

### BuildSlave:

[ubuntu-12\\_04](#)

### Reason:

A build was forced by 'Chris':

### Steps and Logfiles:

- git update ( 3 secs )  
1. stdio
- Configure 'cmake -DCMAKE\_BUILD\_TYPE=Debug ...' ( 8 secs )  
1. stdio
- Compiler compile warnings ( 1 hrs, 34 mins, 28 secs )  
1. stdio  
2. warnings (1)
- Install compile ( 1 mins, 31 secs )  
1. stdio
- Test 'ctest -j2 ...' failed ( 1 hrs, 30 mins, 30 secs )  
1. stdio

### Build Properties:

Name	Value	Source
branch	feature/threading	Build
builddir	/home/buildbot/nektar/slave/ubuntu-12_04-full	slave
buildername	Ubuntu 12.04 64-bit (full)	Builder
buildnumber	659	Build
codebase		Build
got_revision	3176edb8fc5bc697a832c660ff0d64662963566d	Build
log_dir	2	Build
owners	[u'Chris']	Build
project		Build
repository		Build
revision		Scheduler
scheduler	force	BuildSlave
slavename	ubuntu-12_04	BuildSlave
warnings-count	1	WarningCountingShellCommand
workdir	/home/buildbot/nektar/slave/ubuntu-12_04-full	slave (deprecated)

### Forced Build Properties:

Name	Label	Value
force_build_clean	Force a make clean	False
owner		Chris
reason		
url_suffix	Suffix to repo url	nektar/nektar.git
wipe_build	Completely wipe build directory	False

### Responsible Users:

no responsible users

### Timing:

Start: Tue Jun 9 12:34:01 2015  
End: Tue Jun 9 15:40:44 2015  
Elapsed: 3 hrs, 6 mins, 42 secs

### Submit Build:

To force a build, fill out the following fields and push the 'Force Build' button

Reason for re-running build:

Rebuild using:



Resource usage  
CPU usage: 2000m (0.2%)  
Memory usage: 2000m (0.2%)  
Disk usage: 2000m (0.2%)



# Developer Practice

## Builder Ubuntu 12.04 64-bit (full) Build #659

### Results:

#### Failed tests

#### SourceStamp:

Branch feature/threading  
Got Revision 3176edb8fc5bc697a832c660ff0d64662963566d

#### BuildSlave:

ubuntu-12\_04

#### Reason:

A build was forced by 'Chris':

#### Steps and Logfiles:

1. [git update](#)  
1. stdio
2. [Configure 'cmake -DCMAKE\\_BUILD\\_TYPE=Debug ..'](#)  
1. stdio
3. [Compiler compile warnings](#) (1 hrs, 34 mins)  
1. stdio  
2. warnings (1)
4. [Install compile](#) (1 mins)  
1. stdio
5. [Tests 'ctest -j2 ...' failed](#) (1 hrs, 30 mins)  
1. stdio

### Build Properties:

Name	Value	Source
branch	feature/threading	Build
builddir	/home/buildbot/nekta.../slave/ubuntu-12_04-full	slave
buildername	Ubuntu 12.04 64-bit (full)	Builder
buildnumber	659	Build
codebase		Build
got_revision	3176edb8fc5bc697a832c660ff0d64662963566d	Build
revision	2	Build
owners	[u'Chris']	BuildSlave
project		A build was forced by 'Chris':
repository		Build
revision		Build

(view as text)

```
ctest -j2 '--timeout 600' --output-on-failure
in dir /home/buildbot/nekta.../slave/ubuntu-12_04-full/build/builds (timeout 1200 secs)
watching logfiles {}
args: ['ctest', '-j2', '--timeout 600', '--output-on-failure']
environment:
HOME=/home/buildbot
LANG=en_US.UTF-8
LANGUAGE=en_US:en
LOGNAME=buildbot
MAIL=/var/mail/buildbot
PATH=/opt/local/bin:/opt/local/lib/openmpi/bin:/usr/lib/openmpi/bin:/usr/lib64/openmpi/bin:/usr/lib64/mpich/gcc/openmpi/bin
PWD=/home/buildbot/nekta.../slave/ubuntu-12_04-full/build/builds
SHELL=/bin/sh
TERM=linux
USER=buildbot
XDG_SESSION_COOKIE=20422634a5a369277e2004500000003-1423328811.240864-1156323198
using PTY: False
Test project /home/buildbot/nekta.../slave/ubuntu-12_04-full/build/builds
    Start #437: APESolver_AP..._3DPulse_Wkdg_MODIFIED ..... Passed 201.29 sec
    Start #438: APESolver_AP..._3DPulseWall_Wkdg_MODIFIED ..... Passed 349.90 sec
1/463 Test #438: APESolver_AP..._3DPulseWall_Wkdg_MODIFIED ..... Passed 349.90 sec
    Start #163: IncNavierStokesSolver_bfa_tg ..... Passed 152.24 sec
2/463 Test #437: APESolver_AP..._3DPulse_Wkdg_MODIFIED ..... Passed 201.29 sec
    Start #188: IncNavierStokesSolver_Cyl_AdaptiveSFD ..... Passed 123.47 sec
3/463 Test #163: IncNavierStokesSolver_bfa_tg ..... Passed 123.47 sec
    Start #459: MeshConvert_StarTec/StraightRM ..... Passed 113.37 sec
4/463 Test #459: MeshConvert_StarTec/StraightRM ..... Passed 113.37 sec
    Start #202: IncNavierStokesSolver_bfa_tg_par ..... Passed 89.24 sec
5/463 Test #188: IncNavierStokesSolver_Cyl_AdaptiveSFD ..... Passed 89.24 sec
    Start #252: ADRSolver_Advection3D_m12_DQ_hex_VarP ..... Passed 75.62 sec
6/463 Test #252: ADRSolver_Advection3D_m12_DQ_hex_VarP ..... Passed 75.62 sec
    Start #357: ADRSolver_Advection3D_m12_DQ_hex_VarP_par ..... Passed 68.67 sec
7/463 Test #202: IncNavierStokesSolver_bfa_tg_par ..... Passed 68.67 sec
    Start #190: IncNavierStokesSolver_ChainFlow_3DH2D_FFT ..... Passed 39.79 sec
8/463 Test #357: ADRSolver_Advection3D_m12_DQ_hex_VarP ..... Passed 39.79 sec
    Start #148: IncNavierStokesSolver_ChainFlow_3DH2D_MVM ..... Passed 66.52 sec
9/463 Test #190: IncNavierStokesSolver_ChainFlow_3DH2D_FFT ..... Passed 66.52 sec
    Start #144: IncNavierStokesSolver_ChainStability ..... Passed 66.52 sec
10/463 Test #144: IncNavierStokesSolver_ChainStability ..... Passed 66.52 sec
    Start #435: APESolver_AP..._3DPulseWall_Wkdg_MODIFIED ..... Passed 66.52 sec
11/463 Test #148: IncNavierStokesSolver_ChainFlow_3DH2D_MVM ..... Passed 66.52 sec
```

# Developer Practice

What development practices support a large multi-platform collaborative software project such as Nektar++?

- Version-control (Git + GitLab)

<http://gitlab.nektar.info>

```
git clone https://gitlab.nektar.info/nektar/nektar.git
```



- Tests & Continuous Integration (Buildbot)

<http://buildbot.nektar.info>



- Issue tracking (Trac)

<http://www.nektar.info>

- Documentation (PDF + doxygen)

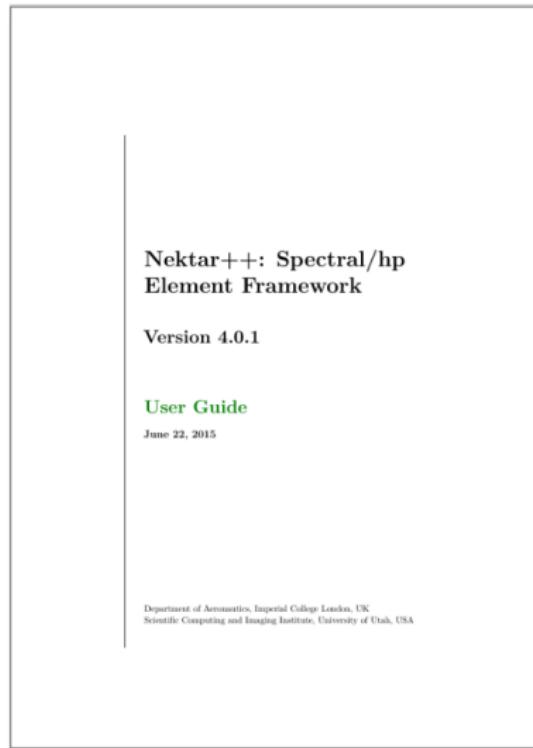
<http://doc.nektar.info>

## Documentation

- All documentation versioned with code

# Documentation

- All documentation versioned with code



- User guide

# Documentation

- All documentation versioned with code

1.3 Installing from Source 5

**Warning**  
Boost version 1.51 has a bug which prevents *Nektar++* working correctly. Please use a newer version.

**Quick Start**  
Open a terminal.  
If you have downloaded the tarball, first unpack it:

```
tar -zxf nektar++-4.0.1.tar.gz
```

Change into the `nektar++` source code directory

```
mkdir -p build && cd build  
cmake ..  
make install
```

**Detailed instructions**  
From a terminal:

1. If you have downloaded the tarball, first unpack it

```
tar -zxf nektar++-4.0.1.tar.gz
```

- User guide
  - Installation

# Documentation

- All documentation versioned with code

112 Chapter 8 Incompressible Navier-Stokes Solver

### 8.6.9 Aortic Blood Flow

The following example demonstrates the application of the incompressible Navier-Stokes solver using the Velocity Correction Scheme algorithm for modelling viscous Newtonian blood flow in a region of a rabbit descending thoracic aorta with intercostal branch pairs. Such studies are necessary to understand the effect local blood flow changes have on cardiovascular diseases such as atherosclerosis.

In the following we will numerically solve for the three dimensional velocity and pressure field for steady boundary conditions. The Reynolds number under consideration is 300, which is physiologically relevant.

**Q** Geometry

**O** The geometry under consideration is a segment of a rabbit descending aorta with two pairs of intercostal arteries branching off. The inlet has a diameter  $D = 3.32\text{mm}$ .

**I**

**C**

**D**

**F**

**R**

**Figure 8.10** Reduced region of rabbit descending thoracic aorta.

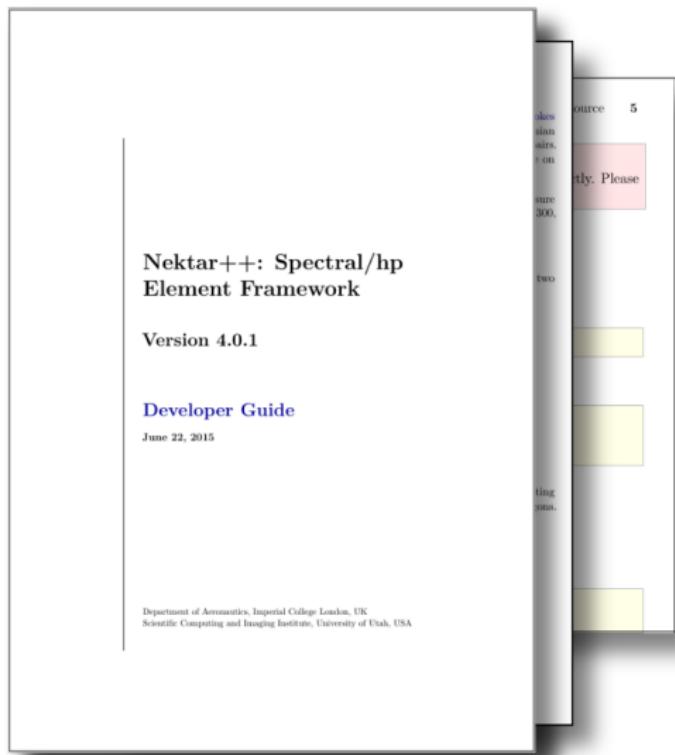
In order to capture the physics of the flow in the boundary layer, a thin layer consisting of prismatic elements is created adjacent to the surface, and curved using spherigons. The interior consist of tetrahedral elements.

**Figure 8.11** Surface mesh indicating curved surface elements at a branch location.

- User guide
  - Installation
  - Solver and Utility usage
  - Tutorials
  - Input file reference
  - FAQs

# Documentation

- All documentation versioned with code



- User guide
  - Installation
  - Solver and Utility usage
  - Tutorials
  - Input file reference
  - FAQs
- Developer guide
  - Programming concepts
  - Library design
  - Nektar++ structures
  - Nektar++ algorithms
  - Coding Standard

# Documentation

- All documentation versioned with code

## Nektar++

Main Page

Related Pages

Namespaces

Classes

Files

 Search

### The Nektar++ library

Nektar++ is an open source software library currently being developed and designed to provide a toolbox of data structures and algorithms which implement the spectral/hp element method. Nektar++ is the continuation and adaptation of the Nektar flow solver. As opposed to its predecessor which focused on solving fluid dynamics problems, Nektar++ is implemented as a C++ object-oriented toolkit which allows developers to implement spectral element solvers for a variety of different engineering problems.

The structure of the Nektar++ library, a collection of different sublibraries, is based upon the typical structure of a global spectral/hp approximation, which is characterized by:

- **The elemental decomposition of the problem**

As for all finite element methods, the computational domain is partitioned into a mesh of many small subdomains or elements. Analogously, the spectral/hp solution is expanded into a series of local expansions, each with support on a single element. This elemental representation enables the treatment of operations on a local elemental basis rather than on global level. This not only simplifies the formulation but also allows many operations to be performed more efficiently.

- **The introduction of a standard region**

The introduction of a standard region allows the expansion basis to be defined just once, that is only on the standard region. All other elements then can be considered as the image of the standard element under a parametric mapping. Consequently, the elemental operations of integration and differentiation can all be executed on the standard element, subject to a proper treatment of the transformation from local (world space) to standard (reference space) coordinates. For curved-sided elements, the mapping from standard element to local element is generally done using an *iso-parametric* representation. In this case, the local geometry is represented with an expansion of the same form and polynomial order as the unknown variables.

This structure, supplemented with building blocks such as block matrix linear algebra routines and automatic data coordinating objects, can be encapsulated in an efficient object-oriented C++ implementation.

This conceptual approach of the software leads to a high user-flexibility, including the selection of the preferred expansion basis, its polynomial order and the preferred numerical quadrature.

The website of the Nektar++ project can be found on: <http://www.nektar.info>

- **User guide**

- Installation
- Solver and Utility usage
- Tutorials
- Input file reference
- FAQs

- **Developer guide**

- Programming concepts
- Library design
- Nektar++ structures
- Nektar++ algorithms
- Coding Standard

- **Code documentation**

- Doxygen
- Detailed implementation specifics

# Summary

- Design of the Nektar++ framework
  - Stack of libraries
  - Code design mirrors mathematical formulation
  - **Encapsulate** complexities of spectral/*hp* element methods
  - Improve **accessibility**
- Collections
  - Multiple implementations for **achieving performance across P**
  - Collates action of operators across **multiple elements**
  - Improves **efficiency** on modern vector-capable CPUs
  - Easy to use with **auto-tuning**
- Developer practice
  - Version control (Git + Gitlab)
  - Regression tests (also useful as examples!)
  - Continuous integration (Buildbot)
  - Issue tracking (Trac)
  - Documentation (LaTeX + Doxygen)

## Further information

- <http://www.nektar.info>
- *Nektar++: An open-source spectral/hp element framework*,  
C. D. Cantwell, D. Moxey, A. Comerford, *et al.*,  
Computer Physics Communications, in press, 2015