

Spectral Element Libraries in Fortran

SELF

Quick Reference Manual

Joseph Schoonover

Contents

Contents	1
1 Software Overview	3
2 Getting Started	5
3 Module List	7
3.1 common	7
3.2 interp	7
3.3 nodal	7
3.4 geom	7
3.5 dgsem	8
3.6 cgsem	8

1 Software Overview

The Spectral Element Libraries in Fortran (SELF) provide supporting data-structures for implementing spectral element methods (SEMs) to solve partial differential equations (PDEs) in multiple dimensions. The focus of the libraries is particularly on Legendre-Galerkin type methods. Legendre-Galerkin methods solve PDEs by approximating the solution and geometry by interpolating polynomials. Discrete equations are formed by solving the weak form of the PDE in which the integrals are approximated by discrete Gauss or Gauss-Lobatto quadrature. Underneath the umbrella of “Legendre-Galerkin” are Continuous Galerkin (CG) and Discontinuous Galerkin (DG) methods. CG methods are primarily used for elliptic or parabolic PDEs while DG methods are focused on hyperbolic systems.

The software is broken into the following components :

1. Commonly used routines and dictionaries,
2. Interpolation and Quadrature,
3. Geometry,
4. Spectral Operator storage structures,
5. Solution storage data structures,
6. Spectral filters,
7. *High end solvers*

This document primarily focuses on the use and modification of the pre-constructed and tested high-end solvers. The first six components all serve as support for computing high

order discrete differentiation on structured or unstructured meshes. Modification of these lower-end support modules is not recommended.

2 Getting Started

The quickest way to get started is to browse through the examples subdirectory and identify a test-case that is appealing to you. The next steps are to compile and execute the example program. This process is now described for the “spot-advection” example for for advection3d highend solver.

3 Module List

This chapter of the documentation provides an overview of each subdirectory underneath the main `src/` directory. Each subdirectory contains, usually, a collection of modules that corresponds to the subdirectory category. Within the modules are data structures and associated subroutines and functions. This pattern of organization applies to every subdirectory, except for `common/`, which supplies an assortment of useful constants and assorted routines that did not fit directly in any of the other subdirectories.

This chapter contains overviews of the data-structures and the accompanying routines. As part of the overview, calling syntax is provided here so that the end-user can make use of these routines. Keep in mind that, template solvers (under `dgsem/` and `cgsem/`) provide a solid “jumping-off” point for a variety of problems; this chapter should help you better understand how these template solvers work so that you can begin making your own modifications.

3.1 common

`ModelPrecision.f90`

Overview: This module sets the precision of floating point data throughout the rest of the software. Definitions are provided for single and double precision floating point arithmetic, and a selector (`prec`) is provided in order to select either choice.

3.2 interp

3.3 nodal

3.4 geom

3.5 dgsem

3.6 cgsem