

SPMODEL Update

~ new module, unit test, libtoolize, Web design ~

Youhei SASAKI

uwabami@gfd-dennou.org
Department of Mathematics, Kyoto University

2011-08-19

Outline

- ① はじめに
- ② New module
- ③ UnitTest, Libtoolize, documentation
- ④ Web design

Outline

- ① 始めに
- ② New module
- ③ UnitTest, Libtoolize, documentation
- ④ Web design

おことわり

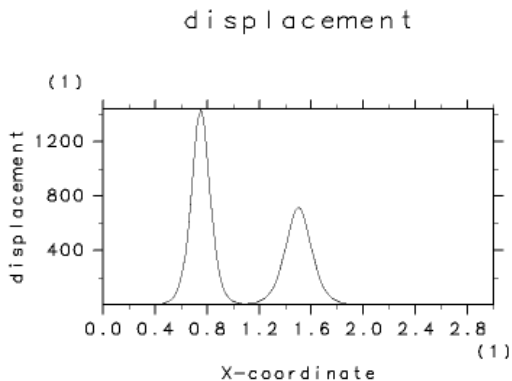
- 主に雑談です.
- 毎度毎度の無保証です.
 - 用法, 用量を守って正しくお使い下さい.
- 疑問, 質問なんでもどうぞ
 - むしろ適宜ツッコんで下さい.

Whats's SPMODEL

SPMODEL :: 階層的地球流体スペクトルモデル集

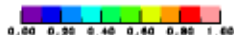
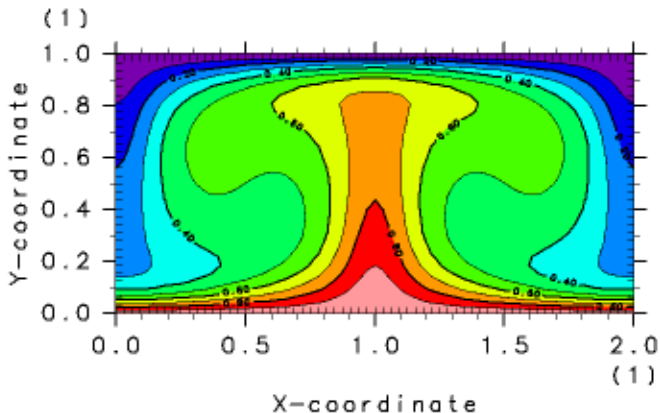
- 地球流体力学に登場するさまざまなレベルの近似方程式系の数値モデルを空間 1 次元 モデルから 2 次元, 3 次元モデルまで階層的に整備.
 - Web: <http://www.gfd-dennou.org/library/spmodel/>
 - ながれマルチメディア:
<http://www.nagare.or.jp/mm/2006/spmodel/>
- 目的
 - 標準的あるいは重要な GFD のイラストレーションをたやすく再現
 - GFD の理解と普及に役立てる
 - 数値計算による知見の共有
 - 従来の「数式」による理解から「数値計算」による理解へ
 - 一連のモデルの振舞いをスムーズに比較
 - より複雑なモデルの結果を理解するための道具

SPMODEL 計算例: KdV ソリトン



SPMODEL 計算例: 二次元ベナール対流

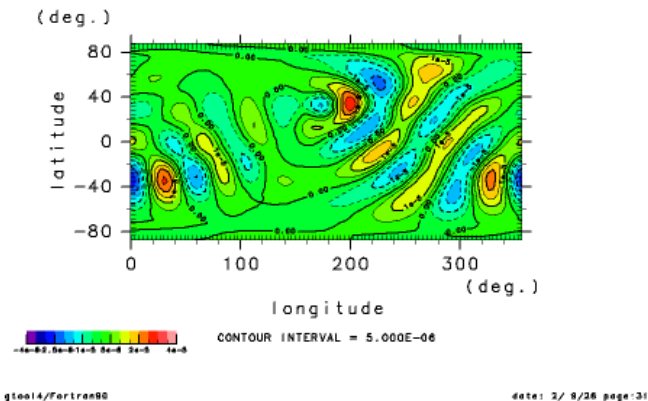
temperature [1]



CONTOUR INTERVAL = 1.000E-01

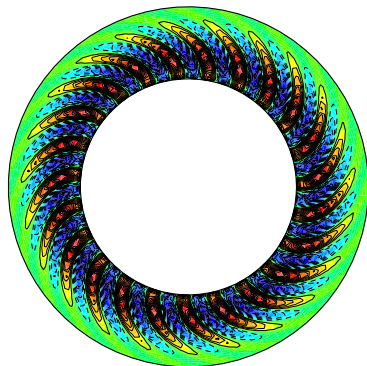
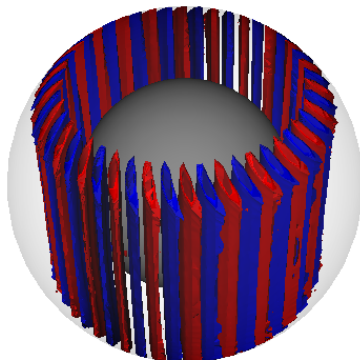
SPMODEL 計算例: 二次元球面

- 山のある浅水系: Gross & Hoskins, 1979 の再計算
vorticity [1/s]



SPMODEL 計算例: 三次元球殻

- 熱対流の臨界モード, 渦度回転軸成分



SPMODEL のコード (抜粋): 一次元移流方程式

使用モジュールの宣言

```
use ae_module
```

変数宣言

```
real(8) :: g_Zeta(0:im-1) ! grid data
real(8) :: e_Zeta(-km:km) ! spectrum data
```

関数群の初期化

```
call ae_Initial(im, km, xmin, xmax)
```

初期値設定

```
g_Zeta = U1 * sech(((g_X-X0)/sqrt(12/u1)))**2 + ...
```

スペクトルに変換

```
e_Zeta = e_g(g_Zeta)
```

時間積分

```
e_Zeta = e_Zeta + dt * ( - c * e_Dx_e(e_Zeta) )
```

出力時は実データに

```
g_Zeta = g_e(e_Zeta)
```

SPMODEL の肝

SPMODEL Library (spml)

- 格子点とスペクトル空間のデータ変換や空間微分などの基本的な配列関数を提供するライブラリ。
 - ISPACK(石岡, 2011) の Fortran (≥ 90) wrapper
- 「テンソルの縮約」のように数式をプログラミング
- 例: KdV 方程式, Leap frog

```
!! leap-frog
e_Zeta = e_Zeta0 + 2.0d0 * dt * (
    - e_g(g_e(e_Zeta1)*g_e(e_Dx_e(e_Zeta1))) &
    - e_Dx_e(e_Dx_e(e_Dx_e(e_Zeta1))) )
e_Zeta0 = e_Zeta1
e_Zeta1 = e_Zeta
...
```

今日のお題: SPMODEL の Update 情報

- 最終リリースが昨年 4 月.
- そろそろまとめてリリースしないといけない(と思う)
- リリースされないプロジェクト = 死んだプロジェクト

Outline

- ① 始めに
- ② New module
- ③ UnitTest, Libtoolize, documentation
- ④ Web design

チェビシェフ-ガウス-ラダウ格子

- 主に二次元極座標, 円筒座標, 球座標での使用を想定
- 原点での特異性を回避しながらスペクトル計算を行なう
- 追加されたモジュール
 - `au_module`: チェビシェフ多項式
 - `aq_module`: ヤコビ多項式
 - `eq_module`: 二次元円盤
 - `wq_module`, `wu_module`: 三次元球
 - `wtq_module`, `wtu_module`: 三次元球殻 + 内球

CUDA 関連

- ISPACK の cuda 関連の関数を呼ぶルーチン
 - `w_module_sjpack_cuda`
 - `wa_module_sjpack_cuda`
 - `wt_module_sjpack_cuda`
- 線形計算に CUDA を使用する関数
 - `lumatrix_cuda`
 - **注意:** 要 PGI Compiler + Accelerator

MPI 関係のルーチン

- ISPACK が MPI 化している関数については, MPI 用の関数群も提供する (予定)
- 追加予定 (作成中)
 - wt_mpi_galerkin
 - wu_mpi_module
 - wq_mpi_module
 - wtu_mpi_module
 - wtq_mpi_module

Outline

- ① はじめに
- ② New module
- ③ Unit Test, Libtoolize, documentation**
- ④ Web design

UnitTest

- spml の提供する全ての関数について, UnitTest を用意する
 - gtool5 内の `dc_test` で整形中
 - 現状, 8 割くらい
- `dc_test` はもっと流行って良い筈.
 - 世にある Fortran の UnitTest framework はどれも貧弱すぎる
 - funit, fruit, ...
- それ以前に Test Driven Development をもっと布教すべき (私が?)
 - Test が無いコード = 「明確な仕様の無いコード」
 - Test を書く事 = コードを文書化すること
 - 「UnitTest」 = 「明確な仕様」 = 「コードの共有化の促進」
 - 「コードを書く時は先ず Test から書く」べき
 - 「〇〇が動かないんだけど...」という不幸な出来事は極力回避したい

Libtoolize

libtool: configure とか Makefile の作成支援

- autotools(autoconf, autoheader, automake) と組み合わせて使う
- 目的
 - ライブラリのポータビリティを上げる
 - Mac だろうと Windows だろうと、動くようにしたい
 - 現状の bashism 溢れる configure をなんとかする
 - 「SunOS では configure が通りません」とか。
- 完成度: 9割できてる
 - Linux, SunOS, AIX, ではテスト済み。
 - SR, SX ではテストしてない(できないので)。
- 将来的には next gtool も libtoolize ... ?

CVS to Subversion

- spmodel 領域の CVS 資源は、全て Subversion へ移行済みです。
- もはや積極的に CVS を使う理由はありません。
 - 今更覚える/思い出す気にはならない
 - 人に教える気もおきない
- せめて Subversion
 - CVS からの移行コストが少ない (ユーザも管理者も)
 - 他の多くのツールも Subversion との連携は (一応) サポートしている
- 将来的には ...?

documentation

- 現状:
 - コード内にドキュメントを埋め込む
 - rdoc-f95 で html 文書化
- 問題点:
 - rdoc-f95 のメンテナンス (引き継いだけど)
 - html 以外の形式での出力?
- 気になっている事
 - Doxygen はどうなんだろう...?

Outline

- ① 始めに
- ② New module
- ③ UnitTest, Libtoolize, documentation
- ④ Web design

RD と仲良くする

- 現状: SPMODEL(に限らず 電脳倶楽部の多くのページ) は RD で記述
 - RD = Ruby 用のドキュメントフォーマット.
 - Perl で言う所の POD
 - 単一のソースに日英併記できる (更新しやすい)
- 問題:
 - 細かい事はできない
 - デザインとは合わない
 - RD に HTML 生書きしている場合
- どうするか?

おしまい

- ありがとうございました
- あとはカメラが止まってから