

dcmode1ミーティング2020年3月26日

ISPACK3について

石岡 圭一 (京大理・地惑)

E-mail: ishioka@gfd-dennou.org

はじめに

ISPACK3は, ispack-3.0.0を昨年3月にリリースして以降, 特に大きな変更はない(ただし, 佐々木さんと竹広さんに見つけていただいたMPIのところのバグ1箇所をfixしたバージョンを昨年末に ispack-3.0.1 としてリリースしている).

今日は, ライバルである SHTns の最新バージョンと比較するためにベンチマークをとりなおしたりしたことについて簡単に報告する.

論文:

Ishioka, K. (2018). A New Recurrence Formula for Efficient Computation of Spherical Harmonic Transform. *JMSJ*, **96**, 241–249.

Legendre陪関数の漸化式にかかる計算コストを従来のものの $1/3$ にする技法。この技法は、他の開発者の球面調和関数変換ライブラリにも既に利用されている。cf. libsharp by Martin Reinecke (<https://gitlab.mpcdf.mpg.de/mtr/libsharp/>)。彼には、libsharpの中に “This update features significant speedups thanks to important algorithmic discoveries by Keiichi Ishioka (https://www.jstage.jst.go.jp/article/jmsj/96/2/96_2018-019/ and personal communication)” と謝辞を書いてもらっている。

他の実装との速度比較

比較対象: SHTns

作者: Nathanaël Schaeffer (ISTerre, Université de Grenoble)

URL: <http://users.isterre.fr/nschaeff/SHTns/>

論文: Schaeffer(2013): Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations, *Geochemistry, Geophysics, Geosystems*, vol. 14, pp.751–758.

開発履歴: Changelog を見ると, 2010年に v1.0 がリリースされていたようだ. 現在は v3.3.1-r694 (2019年9月25日リリース)

実装の方向性は私のものに非常に近い(Legendre陪関数を on-the-fly で計算するとか, SIMD命令の活用とか. 漸化式には特に工夫はない).

ベンチマーク結果

プラットホーム:

CPU: Broadwell Xeon E5-2699v4 (22コア)×2

コンパイラ: ifort 19.0.0.117 -xHost -qopenmp -align array64byte
(SHTns は, gcc 6.3.0, option -O2 -march=native -ffast-math)

44スレッドの場合のみを示す.

SHTns も, v3.2-r660 以降では, “New recurrence formula of Ishioka (2018) leading to faster transforms, especially for large sizes. see <https://doi.org/10.2151/jmsj.2018-019>” として私の手法が取り入れられているので, それとの比較も入れる(ただし, configure 時に `-enable-ishioka` というスイッチをオプションを付けてから make する必要あり).

各スペクトルに $(-1, 1)$ の一様乱数を与え、逆変換 + 正変換を行った誤差評価(SHTns* は, SHTns を `-enable-ishioka` で makeした場合).

		$M = 1023$	$M = 2047$	$M = 4095$
ϵ_{\max}	SHTns	1.8×10^{-12}	6.1×10^{-10}	1.9×10^{-9}
	SHTns*	8.6×10^{-13}	4.0×10^{-12}	2.3×10^{-11}
	ISPACK3	5.1×10^{-13}	1.2×10^{-12}	5.8×10^{-12}
ϵ_{rms}	SHTns	5.7×10^{-14}	2.7×10^{-12}	6.5×10^{-12}
	SHTns*	7.2×10^{-14}	1.8×10^{-13}	4.3×10^{-13}
	ISPACK3	4.3×10^{-14}	8.9×10^{-14}	1.9×10^{-13}

		$M = 8191$	$M = 16383$
ϵ_{\max}	SHTns	5.9×10^{-10}	1.5×10^{-8}
	SHTns*	4.3×10^{-11}	6.1×10^{-11}
	ISPACK3	1.8×10^{-11}	4.1×10^{-11}
ϵ_{rms}	SHTns	1.4×10^{-12}	3.8×10^{-11}
	SHTns*	7.5×10^{-13}	1.3×10^{-12}
	ISPACK3	4.3×10^{-13}	7.9×10^{-13}

スペクトル 格子点値 1回あたりの変換にかかる時間(sec)

	$M = 1023$		$M = 2047$	
	bwd	fwd	bwd	fwd
SHTns	0.0021	0.0025	0.014	0.016
SHTns*	0.0016	0.0018	0.010	0.011
ISPACK3	0.0017	0.0014	0.011	0.0094

	$M = 4095$		$M = 8191$		$M = 16383$	
	bwd	fwd	bwd	fwd	bwd	fwd
SHTns	0.11	0.12	0.87	0.88	7.0	6.8
SHTns*	0.075	0.072	0.56	0.52	4.6	4.0
ISPACK3	0.075	0.064	0.53	0.45	4.0	3.4

スペクトル 格子点値 の変換における実効 GFlops値(このシステムにおける理論ピーク性能は, AVX使用時の CPU周波数を 1.8GHz と仮定すると, $1.8 \times 16 \times 44 = 1267$ GFlops. Turboが効いて, 2.6GHzまで上がるとするならば, $2.6 \times 16 \times 44 = 1830$ GFlops. DGEMMで 4000×4000 と 4000×2000 の行列同士の行列積を計算させて計測すると, 1230GFlops).

	$M = 1023$		$M = 2047$	
	bwd	fwd	bwd	fwd
ISPACK3	651	823	813	940

	$M = 4095$		$M = 8191$		$M = 16383$	
	bwd	fwd	bwd	fwd	bwd	fwd
ISPACK3	933	1090	1052	1245	1111	1311

まとめとTODO

まとめ

- Legendre陪関数の漸化式の工夫など様々な技法をもりこんで, (少なくとも Intelの CPU上では) 世界最速の SHTライブラリを構築した.
- ここで導入した Legendre陪関数の漸化式の工夫については, 他の開発者の実装にも既に取り入れられてきている.

TODO

- 経度方向の回転対称性を仮定した変換の実装
- FFTについて, COS, SIN変換や, 2次元, 3次元のFFTの実装
- Pythonインターフェースの整備