

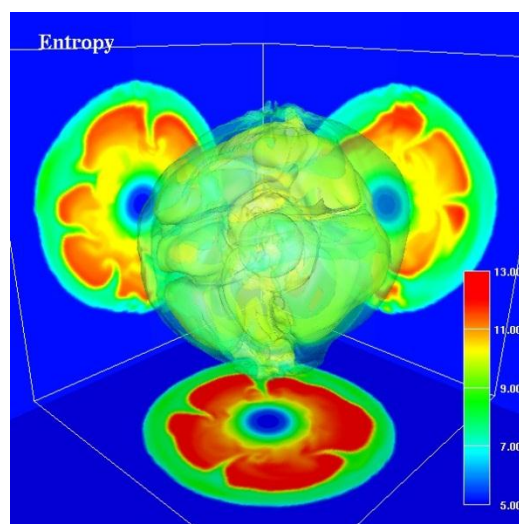
XC30 2013 年 5 月の大規模実行の報告

国立天文台 CfCA 滝脇知也

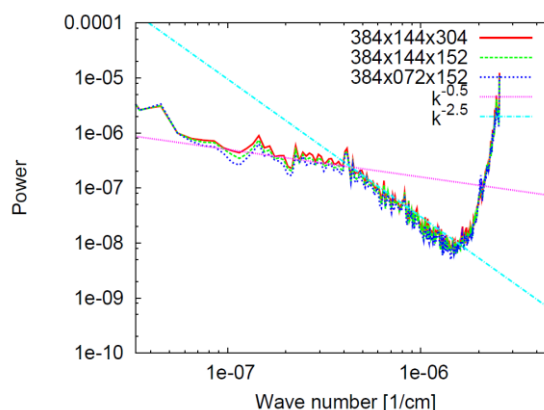
2013 年 5 月 13 日の 17 時から 15 日の 17 時までに行った大規模実行の成果を報告する。

ニュートリノ駆動型超新星のシミュレーションはなかなか成功せず、50 年にも渡って天文学者の頭を悩ませてきたが、近年では成功しつつある。この失敗から成功に転じた大きな理由是对流の取扱いの違いが大きい。旧来は 1 次元球対称の計算をしていたが、近年では計算資源の増加により 2 次元軸対称や 3 次元の計算が行われている。空間の自由度は対流に大きく影響する。まず、1 次元では対流が全く考慮できない。2 次元では対流の効果を考慮できるが形状が正しくない。現実的な状況でニュートリノ駆動による超新星爆発の成否を検証するためには 3 次元シミュレーションが必要である。そこで我々は CfCA の XT4、T2K-tukuba、京コンピュータなどを用いて 3 次元計算を進めてきた。

空間 3 次元の計算はコストが高いため、解像度を高くすることは一般に難しい。今回は XC30 の大規模実行の機会を生かし、超新星爆発の高解像 3 次元計算を行った。今回注目したのは、空間解像度の違いが対流にどのように影響するかである。我々が標準的に用いている解像度は球座標で $384 \times 64 \times 128$ である。この解像度でコアバウンス後 100ms 計算したものを初期条件とした。今回の計算では新たに空間解像度を設定しなおし、3 つのモデルで計算を進めている。モデルの解像度はそれぞれ $384 \times 72 \times 152$, $384 \times 72 \times 304$, $384 \times 144 \times 304$ である。最後のものは昨年度京コンピュータにて計算した高解像度モデル $384 \times 128 \times 256$ より高解像度であり、現時点では世界一の解像度の計算と言える。右上の図は計算開始から 10ms 進んだところのエントロピー分布である。この状態の解析を以下に進める。



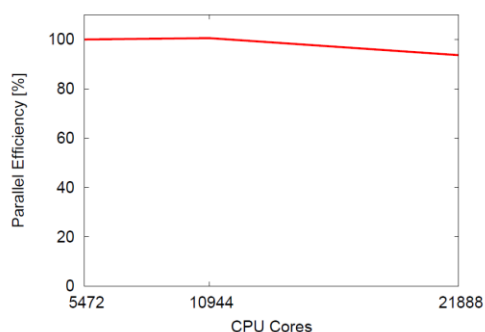
本計算により大質量星内部の対流のスペクトラムが初めて明らかになった。右図は速度から一様に降着する成分を引いた渦の成分をフーリエ分解したパワースペクトラムである。全ての解像度のモデルのパワーが高波数成分（スモールスケール）から低波数成分（ラージスケール）



まで一致し、空間解像度に結果が大きく依らないことが分かる。この程度の空間解像度であれば、信頼のおける結果が得られるということが初めて分かった。

また、非常に興味深いことにラージスケールでは波数の -0.5 乗にパワーが比例し、スモールスケールでは波数の -2.5 乗にパワーが比例した。超新星内部の対流の渦はニュートリノ加熱駆動対流と呼ばれるレイリーテイラータイプの対流と定在降着衝撃波不安定性と呼ばれる流体不安定性が作っていると理解されているが、それらによってできた渦が、お互い合体・消滅した結果、こうしたパワーになることは示されていなかった。なぜこうしたパワーになるのか、なぜ折れ曲がるのかなどは、今後解明しなければならないが、全く分かっていなかった超新星内部の対流構造について一歩踏み込んだ証拠が得られたと言える。

本計算は大規模な並列計算であっても高いスケラビリティを誇っている。右図は今回の計算の並列化効率をグラフにしたもので、横軸はCPUコア数をとっている。今回使用した21888coreはXC30全ノード24192coreの90%以上にあたる。100%の使用を目指したかったが、本計算は空間メッシュ数がある特定の倍数にならなければならなかったゆえ、こうした使用率となった。並列



化効率は21888coreで4分の1の5472coreに対して93%程度となっている。十分高いと判断できるが、Openmpの立ち上げコストが少し見え始めていると考えられる。

本計算の実行時に一番の障害になったのは、ハードディスク容量であった。本計算は1スナップショットあたり、100GBのファイル容量が必要なので200スナップショットでは20TB必要となる。低解像度のモデルはこれよりはファイルサイズが少なくなるが、結局全部で35TB必要であった。ZIPで圧縮すれば3分の1程度になるが、加工するのにも時間がかかるため、瞬間的にはこの程度のファイル容量が必要となる。実際の実行時にはクォタを増加していただいた上でも、ファイルを圧縮しつつ、不必要なファイルを消すなどの作業が求められ、大変苦勞した。大規模実行時にはクォタのサイズをかなり大き目に設定していただくこうした苦勞が少なくなると思われる。

また、実はうっかり計算を7.5時間で打ち切るいつもの設定を消しておらず、14日に24時間流すはずの計算を途中で終わらせてしまった。今回はCrayのミスで他のジョブが消去されており、私のジョブが終了したときに走っているジョブが少なかったため再投入し事なきを得たが、危ないところであった。

最後に CfCA の運用メンバーに感謝して報告を終えたい。本計算は解像度やファイル出力間隔などの、パラメータの設定が直前まで決まらなかったため、クォタの設定を上げてもらう申請などが遅れ、当日の実行は非常に慌ただしくなってしまった。そうした急な要望に応じていただいた石津さんには大変感謝している。