# GPUコンピューティングによる津波解析の高速化と リアルタイム浸水予測

Reatl-time tsunami simulation using GPU

## 越村俊一<sup>1</sup>·香月恒介<sup>2</sup>·茂渡悠介<sup>3</sup>

## Shunichi KOSHIMURA, Kosuke KATSUKI and Yusuke SHIGETO

Using Graphics Processing Unit (GPU), the authors conduct a development of real-time simulation model of tsunami propagation and coastal inundation. As a preliminary result, we terminated the estimation of tsunami travel time and the maximum tsunami height within 8 minutes for 60 minutes of tsunami propagation. The future perspective to develop a tsunami warning system based on the realtime simulation is also discussed.

## 1. はじめに

ゲーム機や画像の描画を目的としてPCに搭載されて いるグラフィックプロセシングユニット(以下GPU)を 科学技術計算にも利用しようとする研究が,GPUコンピ ユーティング技術として注目されている.GPUは1枚あ たり最大240個の微小なプロセッサを搭載し,CPU(現 在は1枚あたり4プロセッサ)の約30倍の処理性能(理 論値)と低コストを誇る.

GPUを用いた津波のリアルタイムシミュレーションに ついては、Acuna and Aoki (2009, 2010) による先駆的 な研究が報告されており、次世代津波予報システムへの 活用が期待されている.本研究では、GPUコンピューテ ィングに特化した津波解析システムを開発し、津波(浸 水域・浸水深・家屋被害)の到達前予測の可能性を検討 することを目的とする.

具体的には、まず、GPU上で高速に動作する津波解析 コードおよび解析システムを開発する.基礎となる解析 プログラムは、非線形長波式をLeap-frog法で離散化した TUNAMI-Codeである.次に、GPU計算による高速化の 評価をおこなう.共通のハードウェアと計算諸元におい てCPU、GPUによる計算をそれぞれ実施して100時間ス テップの計算に必要な時間を計測し、CPUとGPUによる 計算の効率性をそれぞれ詳細に検討する.最後に、GPU 計算によるリアルタイム津波浸水予測の可能性の検討を 行う.東南海・南海地震津波の発生を想定して沿岸各地 における津波到達時間を計算し、現状のハードウェア構 成での津波到達前浸水予測の技術的可能性を論ずる.

1	正会員	博(工)	東北大学准教授 大学院工学研究科
2		修(理)	プロメテック・ソフトウェア株式会社
3		修(理)	プロメテック・ソフトウェア株式会社

### 2. 津波解析コード

まず,GPU上で高速に動作する津波解析コードおよび 解析システムを開発する.図-1にシステム構成を示す. 基礎となる解析プログラムは,非線形長波式をLeap-frog 差分法で離散化したTUNAMI-Code(IUGG/IOC TIME Project, 1997)である.TUNAMI-Codeを基礎プログラム として採用するのは,陽的な差分法であるために並列化 の効率が高いこと,すでにソースコードが公開されてお り多くの国・機関で利用されていること,各種ベンチマ ークにより計算の精度・誤差特性がすでに明らかになっ ていることが理由である(たとえば,Yehら,1996).支 配方程式を式(1)-(7)に示す.

$$\frac{\partial \eta}{\partial t} + \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} = 0 \quad \dots \quad (1)$$

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left(\frac{M^2}{D}\right) + \frac{\partial}{\partial y} \left(\frac{MN}{D}\right) = -gD\frac{\partial \eta}{\partial x} - \frac{\tau_{bx}}{\rho} \quad \dots \quad (2)$$

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left(\frac{MN}{D}\right) + \frac{\partial}{\partial y} \left(\frac{N^2}{D}\right) = -gD\frac{\partial \eta}{\partial y} - \frac{\tau_{by}}{\rho} \quad \dots \quad (3)$$

ここでηは水位, Dは全水深, M, Nは線流量(流量フラ ックス)であり,

となる. また, *x*, *y*方向の底面摩擦項はManningの粗度 係数*n*を用いて次式で表される.

$$\frac{\tau_{by}}{\rho} = \frac{gn^2}{D^{7/3}} N \sqrt{M^2 + N^2} \quad \dots \tag{7}$$



図-1 GPUコンピューティングによる津波解析システムの構成

## 3. 津波解析コードのGPUへの実装

#### (1) CUDAによる実装

NVIDIA社のGPUアーキテクチャ(図-2)では、8個の Scalar Processor (SP) から構成される SIMD型の Streaming Multiprocessor (SM) がGPU上に複数配置され る.SM上では、GPUスレッドは32個ずつのwarpと呼ば れるグループ単位で並列実行される.メモリには全ての スレッドから共有される Global Memory と、SM内のスレ ッド間で共有される Shared Memory があるが、今回の実 装ではShared Memoryは用いなかった.本研究ではGPU にNVIDIA GTX260M (G92b) を採用した. GTX260Mは ノートブックPC用の省電力型GPUで、112個のSPと1GB のメモリを持つ. ホスト側は, CPUに Core2 Duo 2.40GHz を搭載し、メモリを4GB持つノートPCを使用した.

TUNAMI-Codeは、複数サイズの格子をネスティング した場合,主に連続式の計算,開境界処理,領域接続 (水位・流量),運動の式の計算,時刻歴最大値の更新と いった6つ程度の副プログラムで構成される.元々 FORTRANで実装されているTUNAMI-Codeを予めC++ に移植した上で、各プログラムをCUDAのカーネルとし



て実装した.コード保守性の観点から、元のFORTRAN コードからの変更をできるだけ抑えるために、Fortranの 制御構文をC++の構文に置き換えるマクロをいくつか定 義した (リスト-1).

#### a) 二次元配列の更新

いくつかの副プログラムにおいては、ループ内で配列 の各要素が独立に更新される.例えば、二次元配列を更 新するループに対しては、次のような方針で実装を行っ た (リスト-2, 3). まず, 各スレッドは自分のスレッド IDを用いて、インデックスIとJを求める.次に二次元配 列の(I,J)の要素を更新した後、ループの先頭に戻り、 残りの要素を更新する.このカーネルを実行することで, GPUスレッド数(今回は16384個に固定)以上の要素数 をもつ配列を並列に更新している.

さらに、連続式計算のループ内では、配列の同一要素 に対して, 複数のスレッドからの書き込みが同時に発生 し、データ競合が起こる場合がまれにみられた、そのた め,加算代入時に、CUDAの組み込み関数atomicAddを 利用した. atomic 関数を利用すると、メモリ要素の読み 出しと書き込みが排他的に行われ、他のスレッドによっ てデータが上書きされないことが保証される.

また、開境界処理に関しては、更新を行う要素数が GPUスレッド数より小さいため、各スレッドが1要素の 更新を担当するような実装とした.

領域接続については、同様の方針で、単純に FORTRANのループをCUDAのループに置き換えると, データの競合が多数発生し, 正しい更新結果が得られな い.一方,データ競合を防ぐためのatomic 関数は呼び出

リスト-1 FORTRAN風構文マクロの例 (C)

#define $FOR(x,m)$ \	
for $(int x=1;x <=m;x++)$	
$#$ define FOR2(x,b,e) \	
for(int x=b; x <=e; x++)	

リスト-2 二次元配列の更新の例 (C)

#### リスト-3 二次元配列の更新の例 (CUDA)

```
int tidx = blockIdx.x * blockDim.x
    + threadIdx.x;
int step = gridDim.x * blockDim.x;
int tmax = (IF-1)*(JF-1);
//...
for(; tidx<tmax; tidx+=step){
    int J=tidx/(IF-1)+2;
    int I=tidx%(IF-1)+2;
    //...
    VEL(I,J,1)=0.5f*(UU+UUU);
}</pre>
```

しのコストが大きく,多用すると全体のパフォーマンス 低下につながるためここでは不適当である.そこで, JNZ, JNQに関しては,ループの構造を分割し,複数の スレッドが並列実行してもデータ競合が起きないように した.

#### b) PCI-E busのデータ転送の回避

GTX260Mのメモリバンド幅は61GB/sと非常に高いが、 一方で、GPU側のメモリとCPU側のメインメモリの間の データ転送はPCI-Eバス(PCI-Express x16 Gen 2)を介す ため、8GB/sのバンド幅に制限されてしまう。そのため、 GPU-CPU間のデータ転送は最小限に抑える必要がある。 TUNAMI-Codeでは、ループ内でCPUで実行する必要の ある箇所があるとGPUからCPUへ、あるいはCPUから GPUへのデータ転送が頻繁に発生してしまい、処理速度 の低下を招くことになる。そこで、今回のGPU実装では ファイルへの出力を行う副プログラムを除くすべての処 理をGPUに実装し、GPU-CPU間のデータ転送を抑える ようにした。

#### c) Texture cacheの利用

各SM上にはTexture cacheとよばれる読み出し専用の L1キャッシュが存在する.メインループ内で最も処理時 間がかかるのが運動の式の計算で,全体の80%以上を占 めている.そこで,この部分を最適化するために,関数 内で読み出しのみ行われる次の配列に対してTexture cacheを利用した.

## 4. リアルタイム予測に向けての検討

## (1) ハードウェア構成と計算諸元

ここでは、津波の到達前予測を目指したリアルタイム 計算を行うための課題を整理する.表-1に示すハードウ ェアと計算諸元においてCPU、GPUによる計算をそれぞ

表-1 使用したハードウェアと計算諸元

CPU	Intel Core2 Duo P8600 3MB L2 Cash, 2.4GHz
GPU	NVIDIA GeForce GTX 260M
メモリ	メイン:4GB, GPU:1GB
支配方程式	非線形長波式
格子間隔	1350m~50m(4領域)
格子数(領域1)	900×500
格子数(領域2)	1021×601
格子数(領域3)	715×601
格子数(領域4)	1411×1291
時間刻み	0.5 sec.

れ実施して100時間ステップの計算に必要な時間を計測 し、CPUとGPUによる計算の効率性を検討する. また、 ここではテストケースとして東南海・南海地震津波の発 生を想定して沿岸部における津波到達時間を計算し、現 状のハードウェア構成での津波到達前浸水予測の技術的 可能性を論ずる.

#### (2) 津波の波源モデル

シミュレーションの初期条件となる津波発生時の初期 水位分布を津波波源モデルと呼ぶ.津波波源モデルの構 築には地震学の知見(テクトニクス,余震分布,断層メ カニズム,地盤変動量)を考慮し,断層破壊による海底 地盤の変動と等価だと仮定して,くいちがいの弾性論に 基づく解析解(たとえば Okada, 1985)を用いて計算す る.リアルタイム解析を実施する場合には時間的余裕は 無いので,防災科学技術研究所のAQUAシステム(松村 ら,2006)や米国地質調査所(USGS)が配信するCMT 解(Centroid Moment Tensor Solution)を用いるのが一般 的である.特にAQUAシステムでは地震発生から4分か ら8分で震源メカニズム(CMT)が公表されるので,津 波のリアルタイム解析にも活用できる可能性が高い.

#### (3) 検証計算の実施

東南海・南海地震の発生を想定し,津波第1波ピーク と最大波ピークの出現時間との関連で津波到達前浸水予 測の技術的可能性を論ずる.図-4に想定東南海・南海地 震の鉛直地盤変動量分布を示す.図のコンターは地盤の 隆起を実線で,沈降を点線でそれぞれ50cm間隔で表示 した.地盤変動量データは内閣府中央防災会議から提供 を受けた.

計算で得られた津波第1波の出現時間(第1波ピーク 時)の予測結果を図-5に示す.東海から九州にかけての 太平洋沿岸では第一波ピークの出現時間は早くて10分以 内であり,この時間が到達前予測の一つの目安になる. 次に,最大波ピークの出現時間を図-6に示す.太平洋沿 岸では60分以内に最大波が来襲することから,津波第一



図-3 検証計算の領域(空間格子間隔: a:1350m, b:450m, c:150m. d:50m)

副プログラム名	CPUでの実行時間 (ms)	GPUでの実行時間 (ms)	GPU/CPU
水位計算(連続式)	8908.98	345.56	0.039
開境界処理	29.27	4.73	0.162
領域接続(水位)	52.97	39.04	0.737
流量計算 (運動の式)	157497.68	5653.08	0.036
領域接続(流量)	64.76	31.66	0.489
時刻歴最大値の更新	8534.91	110.62	0.013
変数の更新	2812.84	215.63	0.077
合計	177901.41	6400.32	0.036

表-2 津波解析コードの内部処理(100時間ステップ)にかかるCPUとGPUの実行時間の	比輔	Ŋ
--	----	---

波到達と最大波の波高のリアルタイム予測のための現象 再現時間を地震発生から約60分とする.

表-2に示すのは,解析コードを構成する各副プログラ ムの実行時間(100時間ステップ分)をCPU,GPUそれ ぞれに対して求めて比較したものである.従来のCPUを 用いた計算においては,連続式による水位の計算,運動 の式における流量の計算,時刻歴最大値の更新,および 次時間ステップに向けての変数の更新の順に計算負荷が 高く,GPUで演算を行うことでそれぞれ25倍以上の大幅 な高速化を実現することができた.現況のハードウェア 構成においても,60分の現象再現に要する時間は約8分 であり,到達前の予測が可能である.また,6時間分の 津波予測にかかる時間も46分であり,より高速なGPU の並列化や予報区に応じてシステムを分散配置すること で,従来の気象庁のデータベース駆動型津波予報システ ム(舘畑,1996)を圧倒的に凌ぐ精度でリアルタイム浸水 予測が可能であることが分かった.

一方,頻繁に計算結果をファイルに入出力する際には GPUからメインメモリへのアクセスが発生してしまうた め,計算速度は低下する.予報に必要な情報を計算速度 を落とさずに得ることが今後の課題である.

#### 5.まとめ

非線形長波式に基づく津波伝播・浸水予測のシミュレ ーションコードをGPUに実装し,低コスト・高効率の津 波解析システムを開発した.

津波解析コードを構成する各副プログラムの実行時間 をCPU, GPUそれぞれに対して求め, GPU計算の計算効 率を明らかにした.また,解析全体においても約28倍の 高速化を達成することができた.

東南海・南海地震津波の発生を想定した場合,地震発 生から1時間先のリアルタイム予測を行う場合には10分 以内に計算を完了することが可能であり,高知県沿岸・ 紀伊半島の一部の地域を除いて,津波到達前の浸水予測 が可能であることが実証できた.より高速なGPUの並列 化や予報区に応じてシステムを分散配置することで,低 コストかつ高速のリアルタイム津波予報システムの構築 が実現できる.

謝辞:計算の検証に用いた海底地形データおよび津波波 源モデルは内閣府中央防災会議から提供を受けた.本研 究は科学研究費補助金挑戦的萌芽研究(21651078)の補



図-4 想定東南海・南海地震津波の波源(実線は隆起, 点線は沈降, コンター間隔は0.5m)

助を受けて実施された.ここに記して謝意を表する.

#### 参考文献

- 舘畑秀衛 (1998):津波数値計算技術の津波予報への応用,月 刊海洋/号外, No. 15, pp. 23-30.
- 松村 稔,伊藤善宏,木村尚紀,小原一成,関口渉次,堀 貞喜,笠原敬司 (2006):高精度即時震源パラメータ解析 システム (AQUA)の開発,地震2,59, pp.167-184.
- Acuna, M. C. and T. Aoki (2009) : Real-time Tsunami Simulation Accelerated by Parallel GPUs, Proceedings of Computational Engineering Conference, Vol.14, pp.307-310.
- Acuna, M. C. and T. Aoki (2010) : Real-time Tsunami Simulation Accelerated by GPU, Joint Conference Proceedings, 7CUEE & 5ICEE, pp.1701-1707.
- International Oceanographic Commission, IUGG/IOC TIME Project (1997) : Numerical method of tsunami simulation with the leap-frog scheme, *IOC Manuals and Guides*, 35, 126 p.
- Okada, Y. (1985) : Surface deformation due to shear and tensile faults in a half-space, Bull. Seismol. Soc. Am., **75** (4), pp.1135-1154.
- Yeh, H., P. L. F. Liu, and C. Synolakis (eds) (1996) : Long-wave Runup Models, World Scientific, 403 p.



図-5 来襲する津波第1波の出現時間(想定東南海・南海地震)



図-6 来襲する最大波の出現時間(想定東南海・南海地震)