

GPGPU を用いた津波シミュレーションの高速計算

上田 裕貴[†] 古山 彰一[‡]

[†] 富山高等専門学校 〒933-0293 富山県射水市海老江練合 1-2

[‡] 富山高等専門学校 〒933-0293 富山県射水市海老江練合 1-2

E-mail: [†] ue.yuki0206@gmail.com, [‡] shoichi@nc-toyama.ac.jp

あらまし 本研究は、呉羽山断層帯から富山湾沿岸全域に押し寄せ遡上する津波をシミュレーションによって再現し、General-Purpose computing on Graphics Processing Units(GPGPU)を用いて高速化を行うことが目的である。津波の被害を抑えるための対策として、津波の発生情報を元にシミュレーションを行い、津波が浸水する恐れのある地域に警報を発するシステムの構築が期待されている。しかし、富山湾全域かつ 5m 間隔の詳細な地形データを用いてシミュレーションを行うため、計算時間が膨大となる。そこで、多数のコアで並列計算を行い、高い演算性能を発揮する GPGPU を用いてシミュレーション計算の高速化を行う。本研究では GPU を 6 枚用いたシミュレーション計算によって実現象の半分以下の計算時間を実現し、80%以上の並列化効率を得た。

キーワード GPGPU, HPC, 津波, シミュレーション

High speed calculation of tsunami simulation by using GPGPU

Yuki UEDA[†] and Shoichi FURUYAMA[‡]

[†] National institute of technology, Toyama college 1-2 Neria, Ebie, Imizu-shi, Toyama, 933-0293 Japan

[‡] National institute of technology, Toyama college 1-2 Neria, Ebie, Imizu-shi, Toyama, 933-0293 Japan

E-mail: [†] ue.yuki0206@gmail.com, [‡] shoichi@nc-toyama.ac.jp

Abstract A high-speed calculation of the tsunami simulation was realized by using General-Purpose computing on Graphics Processing Unit(GPGPU). In Toyama district, it is concerning that earthquakes occur in the area of Mt. Kureha situated in Toyama Bay, which can then create tsunamis. In this research, the tsunami simulation of Toyama Bay of the actual time phenomena of 200 seconds, was calculated by using the six GPUs. The measured calculation time and did visualization, as a result, the calculation time was 91.8 seconds, and the efficiency of parallel calculation was more than eighty percent. This calculation time is less than half the time of an actual phenomena. And the validity of using GPGPU in calculating simulation was demonstrated by this research.

Keywords GPGPU, HPC, tsunامي, simulation

1. はじめに

本研究では General-Purpose computing on Graphics Processing Units(GPGPU)を用いて、富山湾沿岸で想定される津波の高速な計算と、陸上遡上範囲の予測に応用できる高精度なシミュレーションを実現することが目的である。

富山湾海底には呉羽山断層帯が約 21km に渡って存在する。この断層帯では、将来 M7.4 程度の地震が発生する可能性が示唆されている。断層としては大きいものではないが、沿岸部にかなり近いことから、ここで地震と共に津波が発生した場合には非常に短い時間で沿岸部に津波が到達する事態が懸念される。そのため、津波の発生情報を元に地域に警報を発するシステムの構築が期待されている。本研究では、実際に津波が発生した際に遡上範囲を予測し、被害を軽減するた

めに富山湾全域でのシミュレーションを行う。富山湾全域といった非常に広い範囲でかつ、5m 間隔の詳細な地形データを用いてシミュレーションを行うため、計算時間が膨大となることが予想される。しかし、警報システムに応用するためには津波発生から陸上遡上までの短時間で行える高速な計算が求められる。そこで、大規模な並列計算によって高い演算性能を発揮できる GPGPU を用いて津波の並列計算を行い、シミュレーションの高速化を図る。

2. シミュレーション実験環境

2.1. GPGPU を用いたシミュレーションの高速化

近年、流体シミュレーションの高速化手段の 1 つとして GPGPU が注目されている。GPGPU とは GPU を数値計算目的で使用する手法のことであり、本研究では並列処理で扱いやすい陽解放の津波シミュレーショ

ンを行うため、高速な計算が期待できる GPGPU の使用を試みる。ここでは GPU の特徴を記す。

GPU は主に大量のデータを扱うグラフィックス処理を目的として開発されたハードウェアであり、汎用的な PC では GPU はグラフィックス処理を行い、CPU はシーケンシャルな部分の演算を行うものとして分業している。表 1 は本研究で使用した NVIDIA TESLA K20X の性能を示す。コアが 2688 基搭載されており、最新のプロセッサの CPU でも数個～数十個であるのと比べると非常に多い。この多数のコアを用いて大量のスレッドを並列に動作させることによって CPU 以上の演算性能を引き出すことができる。また TESLA には映像出力機能が無く、GPGPU に特化した構造となっている。

表 1. NVIDIA TESLA K20X の性能

名称	NVIDIA TESLA K20X
コア数[基]	2688
プロセッサ周波数[MHz]	732
単精度演算性能[TFlops]	3.95
メモリバンド幅[GB/s]	250

2.2. 開発環境

本研究では統合開発環境の Compute Unified Device Architecture(CUDA)を利用してシステムの開発を行う。CUDA は GPU の高い演算性能を数値計算に利用することを目的として、NVIDIA 社から提供されている開発環境である。

2.3. 計算モデル

本シミュレーションのコーディングにあたり参照された計算モデルについて説明する。津波のように波長の長い波の運動は浅水理論によって表され、波高 H と波長 L の比 H/L 、及び水深 h と波長の比 h/L がいずれも小さいものとして導かれる。浅水理論を構成する連続の式(1)と運動方程式(2),(3)を以下に示す。

[連続の式]

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

[運動方程式]

$$\frac{\partial M}{\partial t} + \frac{\partial}{\partial x} \left[\frac{M^2}{D} \right] + \frac{\partial}{\partial y} \left[\frac{MN}{D} \right] + gD \frac{\partial \eta}{\partial x} + \frac{gn^2}{D^{7/3}} M \sqrt{M^2 + N^2} = 0 \quad (2)$$

$$\frac{\partial N}{\partial t} + \frac{\partial}{\partial x} \left[\frac{MN}{D} \right] + \frac{\partial}{\partial y} \left[\frac{N^2}{D} \right] + gD \frac{\partial \eta}{\partial y} + \frac{gn^2}{D^{7/3}} N \sqrt{M^2 + N^2} = 0 \quad (3)$$

x, y : 水平方向 h : 静水深 g : 重力加速度

D : 全水深(静水深+水位) η : 静水深からの水位

n : Manning の粗土係数

M, N : x, y 方向の単位幅あたりの流量(流速×全水深)

以上の式を数値的に解くために Leap-frog 法のスキームを用いて離散化し、シミュレーションのコーディングを行う。

2.4. 地形データ

本研究では津波の遡上範囲を正確に予測するために、高解像度な地形データを用いる必要がある。そこで、国土地理院の DEM データ[3]と日本海洋データセンターから提供されている水深データ[4]、船舶からの測定によって座標値と水深が記録されている MGD77 データをもとに作成された 5m 間隔の富山湾全域の地形データを用いてシミュレーションを行う。本研究で使用した地形データを図 1 に示す。図 1 では海域を黒とし、富山湾を囲むように位置する陸域は白黒とした。陸域は黒に近づくほど標高が高いものとする。範囲は縦 15.75km、横 41.625km とする。

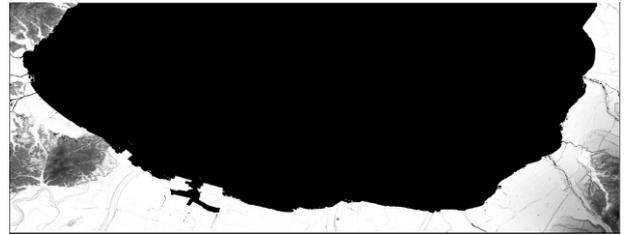


図 2. 使用した地形データ

2.5. 計算環境

今回の実験では GPU1 枚での計算(単一計算)、GPU3 枚での計算(ノード内計算)、GPU6 枚での計算(多ノード計算)を行い、それぞれの計算時間の比較、並列化効率の評価を行う。ここではそれぞれの計算環境について記す。

2.5.1 ノード内計算

ノード内計算における計算環境を図 2、図 3 に示す。GPU を 3 枚搭載したノードを 1 台使用して並列計算を行う。図 3 のように計算領域を均等に 3 分割し、それぞれの GPU で並列計算を行うことによって計算負荷の分散を考える。このように複数枚の GPU を用いて計算を行う場合、隣り合った計算領域を担当する GPU と境界のデータを送受信しながら計算を進めていく必要があるため、その際に発生する時間も考慮する。このように、他の GPU とデータの送受信をし合う際に発生する時間を本研究では境界コピー時間と呼ぶ。また、GPU1 枚のみで計算を行う場合はすべての計算領域を 1 枚で計算するため境界コピー時間は発生しない。

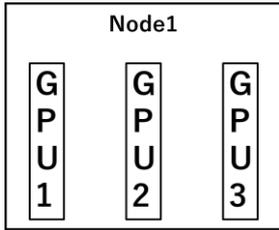


図 2. ノード内計算における計算環境

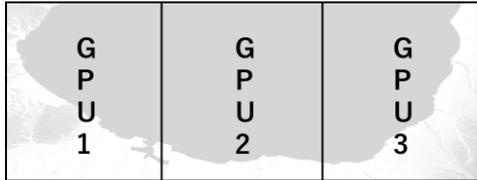


図 3. ノード内計算における計算領域の割り当て

2.5.2 多ノード計算

多ノード計算における計算環境を図 4、図 5 に示す。ノード内計算と同様のノードを 2 台使用し、合計 6 枚の GPU に計算領域を均等に割り当てて並列計算を行う。ノード間をつなぐ LAN は通信速度 40Gbps の InfiniBand を利用する。図 5 に示す GPU3 と GPU4 の境界データの送受信はノード間の通信が必要になる。本研究では Message Passing Interface(MPI)を用いてノード間で境界データを送受信する。

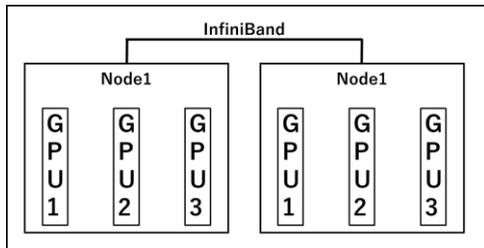


図 4. 多ノード計算における計算環境

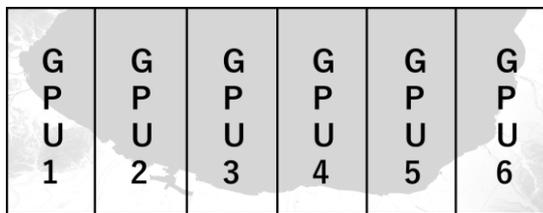


図 5. 多ノード計算における計算領域の割り当て

2.6. フローチャート

シミュレーションのフローチャートを図 6 に示す。最初に、本研究で使用する富山湾の地形データを読み込む。次に津波のパラメータを設定し、初期条件を決定する。この前処理の完了後、津波計算、ファイル出力を指定したステップ数繰り返す。

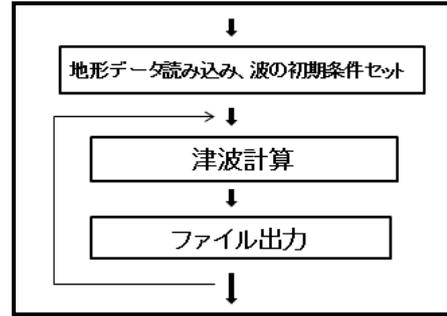


図 6. フローチャート

津波計算は図 7 のように流束計算と水位計算から構成されており、本研究ではこの 2 つの計算部分を GPU による並列計算によって高速化する。また複数の GPU を用いて計算を行う場合は計算領域の境界データを他の GPU に転送する処理（境界コピー）が発生する。津波計算を構成するこれらの処理時間の合計をシミュレーション計算時間として計測する。

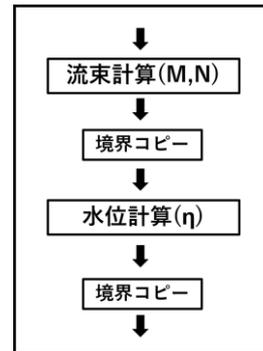


図 7. 計算部分フローチャート

3. 津波シミュレーション条件

シミュレーション評価実験を行うにあたり、シミュレーション条件について記す。計算格子の大きさは 5m の正方形で、格子数は 26,223,750 個(横 8,325×縦 3,150)になる。津波の初期配置は呉羽山断層帯付近(図 8 中、富山湾上の白色の長方形の領域)の水位を 2 秒間かけて 10m 上昇させる。シミュレーションのステップ時間は 0.02 秒で、ステップ数は 10000 回とし、実時間 200 秒の津波シミュレーションを行う。



図 8. 津波の初期配置

4. シミュレーション結果

4.1. シミュレーション可視化結果

津波発生から実時間で 10 秒後、60 秒後、150 秒後の水位、標高を可視化した結果を図 9、図 10、図 11 に示す。図 9 より、呉羽山断層帯付近から波が発生していることがわかる。図 10、11 より、60 秒後には津波が沿岸部に到達し、150 秒後には沿岸地域を遡上していることがわかる。



図 9. シミュレーション結果 (10 秒後)



図 10. シミュレーション結果 (60 秒後)

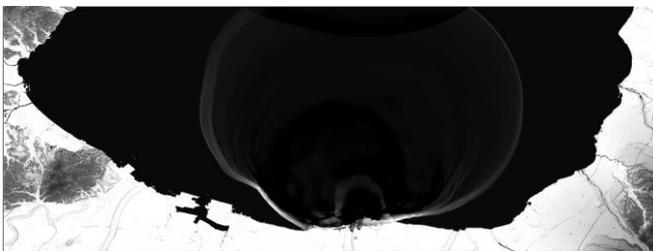


図 11. シミュレーション結果 (150 秒後)

4.2. シミュレーション計算時間

単一計算、ノード内計算、多ノード計算における流束、水位計算時間、境界コピー時間の計測結果と合計計算時間、並列化効率指標を表 2 に示す。流束計算時間、水位計算時間は計算処理を GPU に命令してから全ての GPU が計算処理を終えるまでの時間を計測した。境界コピー時間は水位計算、流束計算終了後から、MPI による通信も含めて境界データのコピーが完了するまでの時間を計測した。これら 3 つを足した合計計算時間がシミュレーションの計算時間となる。並列化効率指標は GPU 枚数を増加させた場合の性能向上効率を表しており、(理想値/実際値)×100 により求めた。ここでの理想値は、GPU3 枚による計算の場合は GPU1 枚の計算時間の 3 分の 1、GPU6 枚による計算の場合は GPU1 枚の計算時間の 6 分の 1 の値とする。

表 2. 計算時間

GPU 枚数	1 枚	3 枚	6 枚
水位計算時間[s]	61.8	20.0	10.2
流束計算時間[s]	410.8	152.2	76.8
境界コピー時間[s]	0.0	2.7	4.9
合計計算時間[s]	472.7	176.1	91.8
並列化効率指標[%]	100	89	86

4.3. 並列化効率についての考察

単一計算(GPU1 枚)と多ノード計算(GPU6 枚)の計算時間を比較する。多ノード計算は単一計算の 6 倍早くなるのが理想であるが、水位計算時間は理想通りの結果を得られたのに対し、流束計算時間は単一計算の 5.38 倍となり、シミュレーション計算の並列化効率を下げていることがわかる。原因としては計算負荷をすべての GPU に均等に分散できていないことが考えられる。水位計算は陸上格子と水上格子の計算負荷は同じであるが、流束計算は陸上格子と水上格子で計算負荷が異なる。陸上格子は隣の格子からの流束がない場合は計算を行わないため、水上格子に比べて計算負荷が少なくなる。今回の実験では計算領域を均等に 6 分割して各 GPU に割り当てたため、陸上格子を多く担当する GPU と水上格子を多く担当する GPU が存在し、その結果計算負荷を均等に分散できなかったと考えられる。流束計算の計算負荷を各 GPU に均等に分散できたと仮定すると、94%の並列化効率が見られると予想できる。

また、GPU 枚数が増えるほど境界コピー時間が長くなり、並列化効率のボトルネックになることが考えられる。高い並列化効率を保つためには、MPI による境界データ送受信中に境界以外の計算領域の計算を行って通信時間の隠蔽を行うといったことが挙げられる。

5. おわりに

本研究では GPGPU を用いて富山湾における実時間 200 秒の津波シミュレーションを行った。GPU6 枚での計算結果は 91.8 秒となり、実現象の半分以下の計算時間を得ることができた。また 80%以上の高い並列化効率を得たことから、GPU を増加させることでシミュレーションの更なる高速化も可能であると考えられる。これらの結果より、実際に津波が発生してから遡上範囲を予測し、被害の及ぶ地域に警報を発するシステムの構築も可能であると考えられる。

謝 辞

本研究は JSPS 科研費 JP16K01339 の助成を受けたものです。

文 献

- [1] 国立大学法人岩手大学, 平成 17 年度宮古湾における津波防災対策検討調査業務委託報告書, pp7-11,14, 2005
- [2] 毎原雄介, GPGPU を用いた富山湾全域における津

波シミュレーションの高速化, pp1-15, 富山高等
専門学校卒業論文, 2015

[3] 国土地理院, "http://www.gsi.go.jp/index.html"

[4] 日本海洋データセンター,
"http://www.jodc.go.jp/service_j.html"