

3 次元 VOF 法に基づく津波シミュレーションのための並列計算

中央大学 学生員 ○ 不室 太希
 東京大学 正会員 田中 聖三
 中央大学 正会員 櫻山 和男

1. はじめに

東北地方太平洋沖地震により発生した津波は、沿岸地域に甚大な被害をもたらし、津波の挙動を事前に予測・把握して災害に備えることの重要性を高めた。

これまでの津波シミュレーションは、広領域を対象として浅水長波方程式や Boussinesq 方程式に基づく 2 次元解析によるものが一般的であるが、構造物への被害を十分に検討するためには Navier-Stokes 方程式に基づく 3 次元解析が必要となる。また、3 次元の津波シミュレーションは非常に大規模な自由度を扱うことから、これを高速に処理するためには並列計算が必要となる。津波の数値解析対象は複雑な自然地形や構造物を考慮するため、任意形状への適合性に優れた有限要素法は有効な手法である。

そこで本研究では、安定化有限要素法¹⁾に基づく 3 次元自由表面流れ解析手法の構築を行うことを目的とし、導入した並列計算手法の効率について検討する。本論文では、MPI, OpenMP および両者のハイブリッド並列計算手法の効率について比較・検討を行う。なお、自由表面位置を捕捉するための計算には、ロバスト性が高い界面捕捉法の 1 つである VOF 法²⁾を用いる。

2. 支配方程式と離散化手法

(1) 支配方程式

非圧縮性粘性流体の流れを考え、流れ場の支配方程式は以下に示す運動方程式 (1) と連続式 (2) で表される。

$$\rho \left(\frac{\partial u_i}{\partial t} + \bar{u}_j \frac{\partial u_i}{\partial x_j} - f_i \right) + \frac{\partial p}{\partial x_i} - \mu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) = 0 \quad (1)$$

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2)$$

ここで、 u は流速、 $\bar{u}$ は移流速度、 p は圧力、 f は物体力、 ρ は密度、 μ は粘性係数を表している。

また、自由表面流れにおける界面位置を表現する VOF 関数は、以下に示す移流方程式によって支配される。

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = 0 \quad (3)$$

ここで ϕ は VOF 関数を表している。

(2) 離散化手法

支配方程式に対して安定化有限要素法を適用し、空間方向の離散化を行う。(1)、(2) 式の弱形式を以下に示す。

$$\begin{aligned} & \int_{\Omega} w_i \rho \left(\frac{\partial u_i}{\partial t} + \bar{u}_j \frac{\partial u_i}{\partial x_j} - f_i \right) d\Omega - \int_{\Omega} \frac{\partial w_i}{\partial x_i} p d\Omega \\ & + \int_{\Omega} \mu \frac{\partial w_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) d\Omega + \int_{\Omega} q \frac{\partial u_i}{\partial x_i} d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \left(\tau_S \bar{u}_k \frac{\partial w_i}{\partial x_k} + \tau_P \frac{\partial q}{\partial x_k} \right) \left\{ \rho \left(\frac{\partial u_i}{\partial t} + \bar{u}_j \frac{\partial u_i}{\partial x_j} - f_i \right) + \frac{\partial p}{\partial x_i} \right\} d\Omega_e \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau_C \frac{\partial w_i}{\partial x_i} \rho \frac{\partial u_j}{\partial x_j} d\Omega_e = \int_{\Gamma} w_i h_i d\Gamma \quad (4) \end{aligned}$$

ここで、 w_i, q は重み関数であり、 τ_S, τ_P, τ_C は安定化パラメータを表す。左辺第 6 項は Shock-Capturing 項である。

(3) 式の弱形式は以下ようになる。

$$\begin{aligned} & \int_{\Omega} w_i \left(\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} \right) d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \left(\tau_{\phi} u_k \frac{\partial w_i}{\partial x_k} \right) \left(\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} \right) d\Omega_e = 0 \quad (5) \end{aligned}$$

ここで、 τ_{ϕ} は安定化パラメータである。

本解析では時間方向の離散化には Crank-Nicolson 法に基づく差分近似を適用する。なお、(4) 式の移流速度 $\bar{u}$ は Adams-Bashforce 法による近似によって計算し、(5) 式の移流速度 u は流れ場の計算から求めたものを使用する。連立一次方程式の解法には GPBi-CG 法を用いる。

3. 並列計算手法と計算機性能

本論文では MPI, OpenMP およびハイブリッド並列計算を行い、プログラムの実行時間の比較を行う。

実行時間の計測に用いる計算機の性能を以下に示す。

〈ハードウェア構成〉

- メモリ：256GB
- CPU：Intel Xeon E5-2687W (3.10GHz, Dual CPU)
- コア数：16 (8 コア × 2)

〈ソフトウェア構成〉

- OS：CentOS5.8
- コンパイラ：Intel Composer XE2011 Linux 版
- 並列化ライブラリ：MPICH2-1.0.8

4. 数値解析例

(1) 解析条件

並列計算手法の違いによる実行時間の比較を行うため、図 - 1 に示すダムブレイク問題を取り上げ、図 - 2 に示す 2 種類のメッシュを用いた解析を行う。微小時間増分量は 0.001[s] とし、1 ステップの実行時間を計測する。なお、気体及び液体の密度はそれぞれ 1.293[kg/m³]、1000.0[kg/m³]、粘性係数はそれぞれ 1.8×10^{-5} [Pa·s]、 1.0×10^{-3} [Pa·s] とする。境界条件として全ての境界面に Slip 条件を与える。

KeyWords： 津波解析, VOF 法, 安定化有限要素法, 並列計算

連絡先： 〒112-8551 東京都文京区春日 1-13-27 E-mail: fumuro@civil.chuo-u.ac.jp

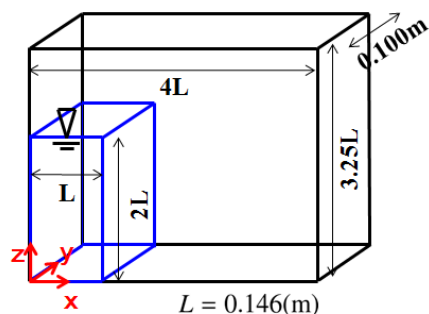


図-1 解析モデル図

メッシュ	最小メッシュ幅[m]	総節点数	総要素数
A	0.005[m]	235872	1322400
B	0.0025[m]	1832454	10624800

図-2 使用するメッシュ

(2) 解析結果

図-3に使用コア数と実行時間の関係を示す。いずれの並列計算手法においても、使用コア数を増やしていくことで実行時間を短縮することができた。同数のコアを使用した場合、OpenMPを用いた計算よりも、MPIを用いた計算の方がより実行時間を短縮できていることが確認できる。また、MPIとハイブリッドを比べた場合、優位性に一貫性が見られないが、これは求解の際の反復回数が異なることが原因として考えられる。

図-4、図-5に使用コア数とスピードアップの関係および使用コア数と並列化効率の関係を示す。ハイブリッド並列計算(MPI:2, OpenMP:8)によって、シングル計算と比べて最大約8倍の速度で計算を行うことができた。使用コア数が増えるにつれてOpenMPの効率がMPIおよびハイブリッドのものより低下しているが、これはメモリアクセスの遅延によるものであると考えられる。本論文で使用した計算機はDual CPUであり、各CPUコアは別のCPUに接続されたメモリに格納されているデータを参照する場合、もう一方のCPUを通してアクセスしなければならない。このため、一ヶ所にデータが格納されているOpenMPでは、メモリアクセスの遅延が生じていると考えられる。これに対してMPIおよびハイブリッドでは、データがメモリ上に分散されるため、OpenMPのようなメモリアクセスの遅延が抑えられている。また、(a)、(b)のグラフを比較することで、各手法において、規模の大きい問題の方が高い並列化効率を得ることが確認できる。

5. おわりに

本報告では、3次元VOF法に基づく大規模津波シミュレーションのための並列計算法を構築するため、MPI、OpenMP、および両者のハイブリッド並列計算による実行時間を計測することで以下の結論を得た。

- 同じ使用コア数で並列計算手法を比較した場合、MPIまたはハイブリッド並列計算は、OpenMPを用いる並列計算より実行時間が短縮されることが明らかとなった。

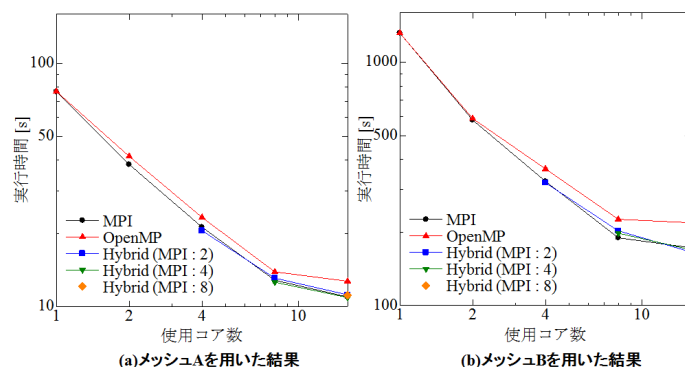


図-3 使用コア数と実行時間の関係

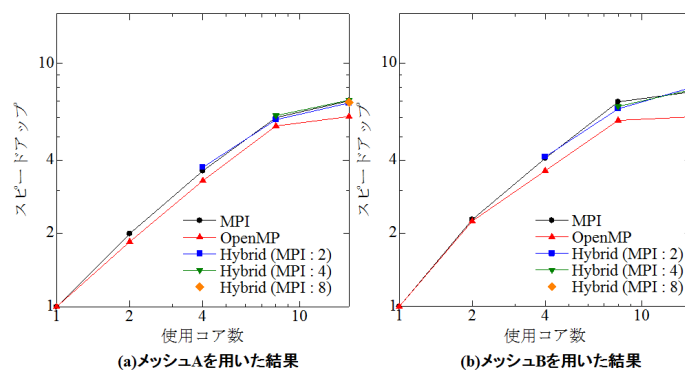


図-4 使用コア数とスピードアップの関係

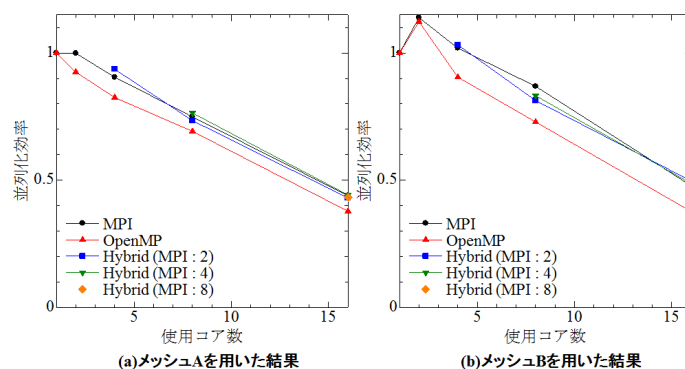


図-5 使用コア数と並列化効率の関係

- 本解析で用いた計算機において、MPIとOpenMPとでは、使用コア数を増やしていくにつれてMPIの優位性が顕著に現れた。
- 各手法において、規模の大きい問題の方が高い並列化効率を得ることが明らかとなった。

今後の方針として、さらなる精度検証を行っていくとともに、都市域における3次元大規模津波シミュレーションを行っていく予定である。

参考文献

- 1) T.E.Tezduyar: Stabilized finite element formulations for incompressible flow computations, *Advance in Applied Mechanics*, 28, pp.1-44, 1992.
- 2) C. W. Hirt and B. D. Nichols: Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *Journal of computational physics*, 39, pp.201-225, 1981.
- 3) 日本計算工学会流れの有限要素法研究委員会: 続・有限要素法による流れのシミュレーション, シュプリンガー・ジャパン, 2008.
- 4) 櫻山和男, 西村直志, 牛島省: 並列計算入門, 丸善, 2003