境界要素法による大振幅水面波の数値計算

秋田大学理工学部　平川知明
Tomoaki Hirakawa
Faculty of Engineering Science,
Akita University

1　研究目的

持続可能な社会の実現・地方創生という観点から，波力発電（WEC）や洋上風力発電が期待されている．WEC に限らず浮体の揺動シミュレーションのために活発に研究開発されてきた計算手法の 1 つとして，直接境界要素法（パネル法）がある．この方法は，計算精度が比較的高いことで知られており，また，3 次元の問題を 2 次元の問題に変換するため，解くべき連立 1 次方程式の数を格段に減らすことができ，この点においては他の計算手法と比べて優位である．しかし，一方で，連立 1 次方程式を作ることに膨大な計算時間を要してしまい，直接境界要素法は大規模シミュレーションには不向きと考えられているようである．

このボトルネックを改善するため，最近では Fochesato and Dias (2006) が水面波シミュレーションにおいて，Fast multipole method に用いることで，6022 節点の場合に 6 倍程度速く計算することに成功している．また，並列計算による高速化も改善に有効な手段として考えられる．

他にも直接境界要素法の課題として，時間発展に伴い水面の補間精度が悪化することが挙げられる．補間精度を向上させる様々な方法が提案されてきたが，確固たる方法は未だないように思われる (Otta, Svendsen & Grilli, Otta et al.; Grilli and Svendsen, 1990; Seo and Dalrymple, 1990; Grilli and Subramanya, 1996)．

本研究では，安定した浮体揺動シミュレーションプログラムを開発する前段階として，3 次元大振幅水面波シミュレーション効率の改善を目的としている．

2　直接境界要素法（パネル法）

2.1　境界積分方程式の離散化

水面波のシミュレーションのために，非粘性非圧縮渦なし流れを仮定し，速度ポテンシャル ϕ を扱う．計算する境界面上の節点では，ϕ または，その法線方向微分 ϕ_n が初期値としてわかっている必要がある．流体全体で成り立つラプラス方程式 $\Delta\phi = 0$ は，グリーンの定理：

$$\iiint_V (\phi \Delta G - G \Delta \phi)\, dV = \iint_S (\phi \nabla G - G \nabla \phi) \cdot \mathbf{n} dS, \tag{1}$$

と合わせて，$G = 1/|\mathbf{x} - \mathbf{a}|$ として，次の境界積分方程式で置き換える．

$$c\phi(\mathbf{a}) = \iint_S (\phi \nabla G - G \nabla \phi) \cdot \mathbf{n} dS. \tag{2}$$

滑らかな境界面では定数は $c = -2\pi$ となる．

境界面 S を，ディリクレ，ノイマンの境界条件別に分割した後，それら各境界面をさらに細かく要素で分割する．要素とは，境界面上のいくつかの節点からなるパラメトリック補間のことで，(2) の境界積分方程式は，この補間を適用することで離散化できる．例えば，境界 i の要素 j にお

いて節点が格子状に与えられており，各節点 $\mathbf{k}$ における速度ポテンシャルを $\phi^{(i,j,\mathbf{k})}$ とすると，パラメトリック補間 $\phi^{(i,j)}(\xi,\eta)$ は次のように表すことができる．

$$\begin{cases} \mathbf{x}^{(i,j)}(\xi,\eta) = \displaystyle\sum_{k_1=0}\sum_{k_2=0} \mathbf{x}^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})}(\xi,\eta) \\ \phi^{(i,j)}(\xi,\eta) = \displaystyle\sum_{k_1=0}\sum_{k_2=0} \phi^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})}(\xi,\eta) \\ {\phi_n}^{(i,j)}(\xi,\eta) = \displaystyle\sum_{k_1=0}\sum_{k_2=0} {\phi_n}^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})}(\xi,\eta) \end{cases}. \tag{3}$$

ここでの N は形状関数である．境界 i の要素 j において G，∇G を (3) で補間すると

$$\begin{aligned} G^{(i,j)}(\xi,\eta;\mathbf{a}) &= \frac{1}{\left|\displaystyle\sum_k \mathbf{x}^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})} - \mathbf{a}\right|}, \\ (\nabla G)^{(i,j)}(\xi,\eta;\mathbf{a}) &= -\frac{\displaystyle\sum_k \mathbf{x}^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})}(\xi,\eta) - \mathbf{a}}{\left|\displaystyle\sum_k \mathbf{x}^{(i,j,\mathbf{k})} N^{(i,j,\mathbf{k})}(\xi,\eta) - \mathbf{a}\right|^3}. \end{aligned} \tag{4}$$

(3) と (4) を (2) に代入すると，(2) は ϕ と ${\phi_n}$ の 1 次式方程式で表される：

$$\begin{aligned} c\phi(\mathbf{a}) =& \sum_{i=0}^{\mathrm{n_b}}\sum_{j=0}^{\mathrm{n_e}}\sum_{\mathbf{k}=(0,0)}^{\mathbf{n_n}} \phi^{(i,j,\mathbf{k})} \sum_{m,n=0}^{\mathrm{n_g}} (\nabla G)^{(i,j)} \cdot \left(\frac{\partial \mathbf{x}^{(i,j)}}{\partial\xi} \times \frac{\partial \mathbf{x}^{(i,j)}}{\partial\eta}\right) N^{(i,j,\mathbf{k})} w_{m,n} \\ &- \sum_{i=0}^{\mathrm{n_b}}\sum_{j=0}^{\mathrm{n_e}}\sum_{\mathbf{k}=(0,0)}^{\mathbf{n_n}} {\phi_n}^{(i,j,\mathbf{k})} \sum_{m,n=0}^{\mathrm{n_g}} G^{(i,j)} N^{(i,j,\mathbf{k})} \left|\frac{\partial \mathbf{x}^{(i,j)}}{\partial\xi} \times \frac{\partial \mathbf{x}^{(i,j)}}{\partial\eta}\right| w_{m,n}. \end{aligned} \tag{5}$$

積分をガウス求積で数値積分するため重み w が含まれている．

(5) の $\mathbf{a}$ に，全節点の位置を代入すれば，十分な大きさの連立 1 次方程式が得られる．この方程式を解くことで，ϕ が与えられているディリクレ境界条件の節点には ${\phi_n}$ が，${\phi_n}$ が与えられているノイマン境界条件の節点には ϕ が得られ，境界面上の全節点で $(\phi, {\phi_n})$ が得られる．

(2) を補間によって (5) のように離散化する過程において，境界を区別する添え字 i と要素を区別する添え字 j があると便利である．

2.2 MEL を使った時間発展

流体粒子の位置とそこでの速度ポテンシャル ϕ は，Mixed Euler-Lagrange(MEL) 法でラグランジュ的に時間発展させる．水面における ϕ のラグランジュ微分は，$p=0$ なので $\nabla\phi$ がわかれば次式から計算でき，$\partial\phi/\partial t$ を計算する必要がない．

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \nabla\phi\cdot\nabla\phi = \frac{1}{2}(\nabla\phi\cdot\nabla\phi) - gz - \frac{p}{\rho g}. \tag{6}$$

(5) を解くことで，境界面上の全ての点において ϕ，${\phi_n}$ が分かっている．残りの接線方向 (s,m) の速度ベクトル ϕ_s，ϕ_m は，パラメトリック補間の微分を使って次式で計算できる．

$$(\phi_s, \phi_m) = \left(\frac{\partial\xi}{\partial s}\frac{\partial\phi}{\partial\xi}, \frac{\partial\eta}{\partial m}\frac{\partial\phi}{\partial\eta}\right). \tag{7}$$

(7) は，境界 i の要素 j においては次式を使って計算できる．

$$\left(\frac{\partial\phi}{\partial\xi}^{(i,j)},\frac{\partial\phi}{\partial\eta}^{(i,j)}\right)=\left(\sum_{\mathbf{k}}\phi^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\eta},\sum_{\mathbf{k}}\phi^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\xi}\right),$$
$$\begin{cases}\frac{ds}{d\xi}^{(i,j)}(\xi)=\sqrt{\left(\sum_{\mathbf{k}}x^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\xi}\right)^2+\left(\sum_{\mathbf{k}}y^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\xi}\right)^2+\left(\sum_{\mathbf{k}}z^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\xi}\right)^2}=\left|\frac{d\mathbf{x}}{d\xi}^{(i,j)}\right|\\ \frac{dm}{d\eta}^{(i,j)}(\eta)=\sqrt{\left(\sum_{\mathbf{k}}x^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\eta}\right)^2+\left(\sum_{\mathbf{k}}y^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\eta}\right)^2+\left(\sum_{\mathbf{k}}z^{(i,j,\mathbf{k})}\frac{dN^{(i,j,\mathbf{k})}}{d\eta}\right)^2}=\left|\frac{d\mathbf{x}}{d\eta}^{(i,j)}\right|\end{cases}. \tag{8}$$

以上の方法で得られたローカルな流速ベクトル $\nabla_L\phi=(\phi_s,\phi_m,\phi_n)$ は，グローバルな流速ベクトルへと次式で変換する．

$$\nabla\phi=\mathbf{M}_{LtoG}\cdot\nabla_L\phi,\quad \mathbf{M}_{LtoG}=\begin{bmatrix}|(x_\xi,y_\xi,z_\xi)|^{-1}(x_\xi,y_\xi,z_\xi)\\ |(x_\eta,y_\eta,z_\eta)|^{-1}(x_\eta,y_\eta,z_\eta)\\ |(x_\xi,y_\xi,z_\xi)\times(x_\eta,y_\eta,z_\eta)|^{-1}(x_\xi,y_\xi,z_\xi)\times(x_\eta,y_\eta,z_\eta)\end{bmatrix}^{-1}. \tag{9}$$

これまでの研究では，比較的少ない数の節点からなる補間が用いられてきた．この方法は，補間に費やす計算コストが安く済むが，一方で，補間端点が境界面上に無数に存在することになるため，境界面の補間精度が悪化する可能性がある．そこで，本研究では，B スプラインを用いて，できる限り補間の端点ができないように境界面を補間し，これに伴う計算コスト増加は，OpenMP による並列計算と数値積分の効率を向上させることで抑えることにした．

ある変数 s の (ξ,η) のパラメトリック補間は，B スプライン基底 B を用いて次式のようにテンソル積で表される．2 重和の上限はサンプル点の数で決まり，B スプライン基底の次数 K,J は任意に決めることができる．

$$s(\xi,\eta)=\sum_{k_1=1}\sum_{k_2=1}a_{k_1,k_2}B_{k_1,K}(\xi)B_{k_2,J}(\eta). \tag{10}$$

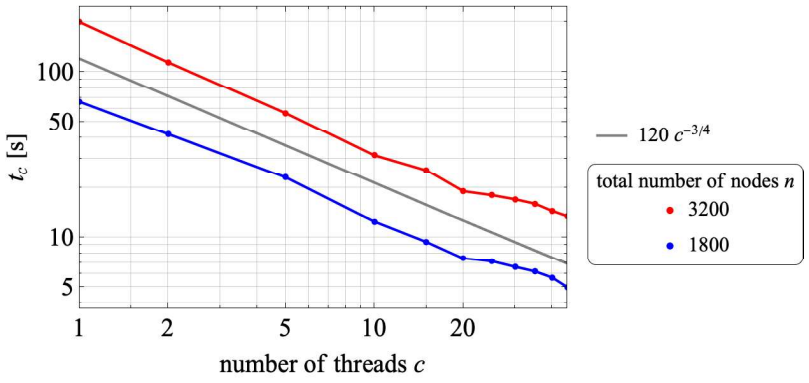
係数 $a_{i,j}$ は，与えられた境界面上の節点の情報から，ステップ毎に計算する．

3　計算効率改善の結果と考察

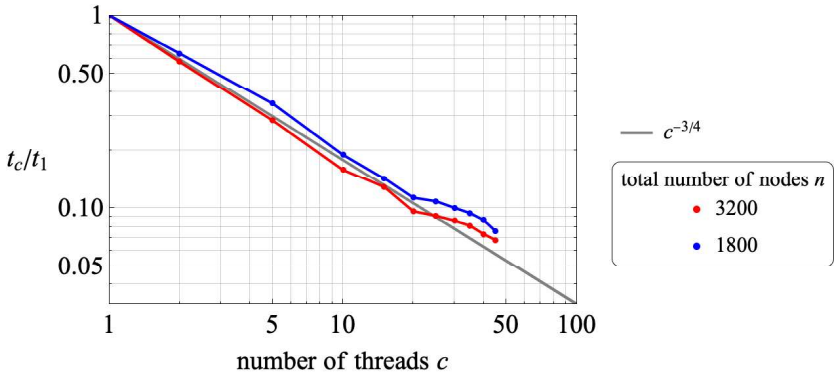
3.1　並列計算

(2) の $\mathbf{a}$ を各節点の位置として 1 次方程式を作成していくことで，問題を解くのに十分な大きさの連立 1 次方程式を得ることができる．並列化はこの異なる節点位置 $\mathbf{a}$ に関して行う．

境界面を水底と水面に分け，全節点数が 3200，1800 の場合の，並列計算の効率を調べた（図 1(a)）．スレッド数が 20 以下の場合は，OpenMP による計算時間の短縮率は $c^{-3/4}$ で近似できる（AMD Ryzen Threadripper 2970WX 24-Core Processor を使用）．つまり，スレッド数が 20 あれば，シングルスレッドの場合と比べて約 1/10 の時間で計算できる．また，全節点数 n が大きい，大規模な計算になる程，並列計算の効率は良くなる．しかし，スレッド数が 20 を過ぎると時間短縮は鈍化する傾向があり，さらに効率を上げるには，数値積分や補間の効率化が必要と考えられる．



(a) 実際の計算時間



(b) シングルスレッドの計算時間で規格化

図 1: OpenMP による計算時間の短縮率．シングルスレッドの計算時間を t_1，スレッド数 c の計算時間を t_c とする．

3.2 数値積分

パラメトリック補間のパラメタ (ξ, η) の範囲は，簡単のために $-1 \leq \xi \leq 1, -1 \leq \eta \leq 1$ として，数値積分にはガウスルジャンドル求積を用いる．例えば，G の η に関する積分は，次のようにガウス点 $\{\eta_0, \eta_1, ..., \eta_N\}$ とそこでの重み $\{w_0, w_1, ..., w_N\}$ を使って計算する．

$$\int_{-1}^{1} G(\eta)\, d\eta = \sum_{i=0}^{N} G(\eta_i)\, w_i \tag{11}$$

積分される関数の特異点付近で，ガウス点を密に取るように変数変換すれば積分精度は向上するので，次のように変数変換する．ここで β はガウス点の集中度合を調整するパラメタになっている．

$$\eta = f(u) = \eta_{\rm sig} + \mathrm{sgn}(u)\,|u|^{\beta}, \quad u = f^{-1}(\eta) = \mathrm{sgn}(\eta - \eta_{\rm sig})\,|\eta - \eta_{\rm sig}|^{1/\beta}. \tag{12}$$

まず，$f^{-1}(\eta = -1) \leq u \leq f^{-1}(\eta = 1)$ でガウス点 $\{u_0, u_1, ..., u_N\}$ とその重み $\{w_0, w_1, ..., w_N\}$ を計算し，次に，$\eta = f(u)$ を使って，ガウス点は η 空間に戻し，重みには $df(u)/du$ をかける．

$$\int_{u=f^{-1}(-1)}^{u=f^{-1}(1)} G(f(u)) \frac{df(u)}{du} du = \sum_{i=0}^{N} G(f(u_i)) \frac{df(u)}{du} w_i \tag{13}$$

表 1，2 は，以上の方法による，$\int_0^1 \log(x)\,dx + 1$ と $\int_0^1 x^{-1/2}dx - 2$ の数値積分結果である．ガウス点の集中度を表す β が大きくなるにつれて，積分される関数の特異的な変化を捉えられるようになり，積分精度が向上していることがわかる．特に，$\int_0^1 x^{-1/2}dx - 2$ の場合は，積分される関数が冪関数なので，β が 2 の倍数で特異性がなくなる．

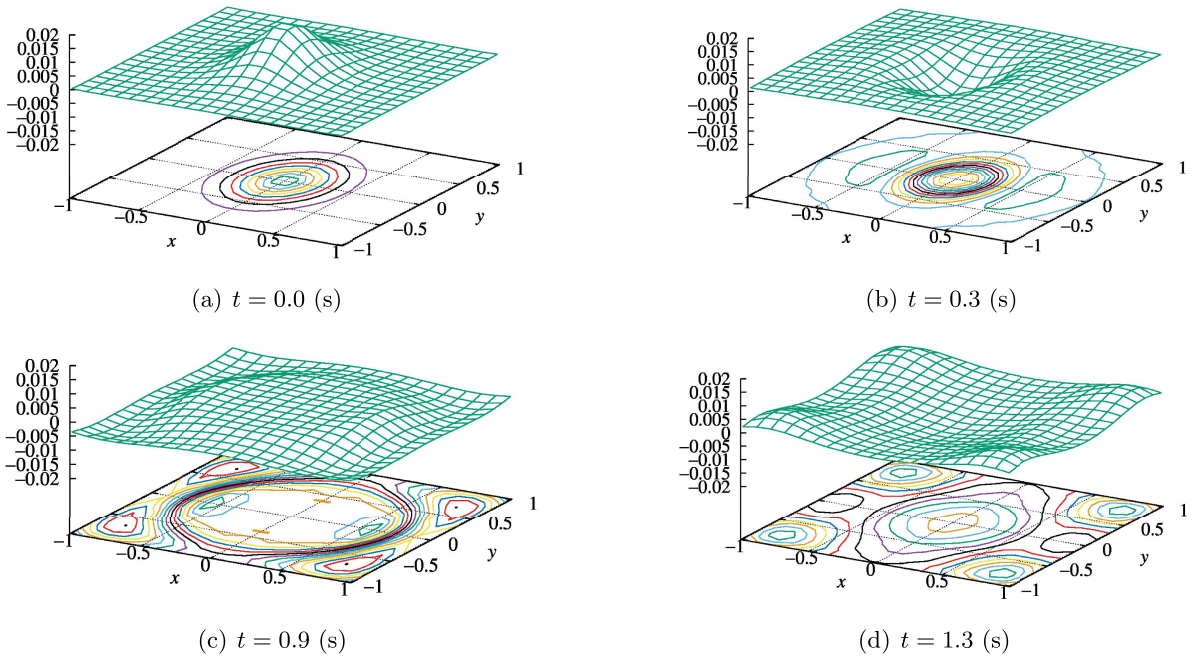
表 1: $\int_0^1 \log(x)\,dx + 1 = 0$

	β					
number of Gauss points	1	1.5	2.0	2.5	3.5	4.0
20	1.50e-03	-1.86e-04	-5.70e-06	8.21e-07	-6.59e-09	-4.68e-10
30	6.79e-04	-6.28e-05	-1.16e-06	1.28e-07	-4.70e-10	-1.93e-11
40	3.85e-04	-2.88e-05	-3.74e-07	3.36e-08	-7.07e-11	-1.99e-12

表 2: $\int_0^1 x^{-1/2}dx - 2 = [2\sqrt{x}]_0^1 - 2 = 0$

	β					
number of Gauss points	1	1.5	2.0	2.5	3.5	4.0
20	-4.25e-02	-4.12e-03	-4.44e-15	1.25e-04	5.53e-06	-4.00e-15
30	-2.85e-02	-2.27e-03	-6.22e-15	4.64e-05	1.38e-06	-6.66e-15
40	-2.15e-02	-1.48e-03	-1.82e-14	2.28e-05	5.10e-07	-1.80e-14

以上の方法で数値シミュレーションした例が表 3.2 である．時間発展は 4 次のルンゲクッタを用いて，流体粒子は Mixed Euler-Lagrange(MEL) を使った．また，数値積分におけるガウス点は 40 点，β は 2.0 に設定した．



(a) $t = 0.0$ (s)　(b) $t = 0.3$ (s)

(c) $t = 0.9$ (s)　(d) $t = 1.3$ (s)

図 2: 3 次元水面波シミュレーションの計算結果例

4　まとめと今後の課題

浮体動揺シミュレーションの前段階として，３次元大振幅水面波シミュレーションのためのプログラム開発および，並列計算と数値積分改善による効率化を行ったが，３次元大振幅水面波の計算までには至らなかった．

OpenMP による並列計算は簡単であり，劇的に時間を短縮できるため有用であるが，全節点数が数万に及ぶような大規模シミュレーションを行うには，これだけでは十分に時間を短縮できない．

ガウス点を特異点付近に集中させるよう変数変換することで，積分精度が向上し演算回数を減らすことができる．水面形状がシンプルな場合は，この方法は有用であると思われるが，水面形状が複雑な場合は精度が悪化することも考えられる．この点においてさらに調べる必要がある．

参考文献

Fochesato, C. and F. Dias, 2006: A fast method for nonlinear three-dimensional free-surface waves. *Proc. R. Soc. A Math. Phys. Eng. Sci.*, **462**(2073), 2715–2735.

Grilli, S. T. and R. Subramanya, 1996: Numerical modeling of wave breaking induced by fixed or moving boundaries. *Comput. Mech.*, **17**(6), 374–391.

Grilli, S. T. and I. A. Svendsen, 1990: Corner problems and global accuracy in the boundary element solution of nonlinear wave flows. *Eng. Anal. Bound. Elem.*, **7**(4), 178–195.

Otta, A. K., I. A. Svendsen, and S. T. Grilli Unsteady Free Surface Waves in a Region of Arbitrary Shape. *Res. Rep. No CACR Cent. Appl. Coast. Res. Univ Delaware 153 pp* 0–92.

Seo, S. N. and R. A. Dalrymple, 1990: An efficient model for periodic overturning waves. *Eng. Anal. Bound. Elem.*, **7**(4), 196–204.