勾配法による構造物の釣合解析に関する数値実験
Numerical experiments on equilibrium analysis of structures by gradient methods

○藤井　文人[*1]，藤田　慎之輔[*2]
Fumito Fujii[*1]，Shinnosuke Fujita[*2]

*1 北九州市立大学 国際環境工学部 学士課程
Under Graduate School, Faculty of Environmental Engineering, The University of Kitakyushu
*2 北九州市立大学大学院 国際環境工学研究科 講師 工博
Lecturer, Graduate School of Environmental Engineering, The University of Kitakyushu, Dr. Eng.

キーワード：全ポテンシャルエネルギー最小化　最急降下法　加速勾配法　共役勾配法
Keywords：Total potential energy minimization　steepest descent method　accelerated gradient method　conjugate gradient method

1. はじめに

近年，益々形態が複雑化する建築構造物に対して，構造物の実挙動を正確に把握するためには，大規模な解析モデルを作成する必要があり，数値解析に膨大な計算負荷を伴うこととなる．解析モデルが大規模化することによる数値解析時間の肥大化は設計過程のトライアル・アンド・エラーの機会を奪ってしまうことにつながるため，従来の方法に代わる高速な数値計算アルゴリズムの構築が望まれている．

近年，数学の世界において，大規模な最適化問題を解く効果的な解法として，適応再スタート付き加速付き勾配法 (以下，加速勾配法) が注目を集めている．同法は，例えばビッグデータに対する機械学習などに実用化されており [1]，建築構造分野への拡張も可能と考えられる．既往の研究では，トラス構造物の材料非線形解析や，平面骨組の幾何学的非線形解析に対して，剛性方程式を解く代わりに全ポテンシャルエネルギー最小化問題を加速勾配法を用いて解く数値実験が行われ，一定の有効性が示されている [2–5]．

本報告では，加速勾配法を含む複数の勾配法について，各最適化ステップにおけるステップ幅の設定方法を様々に変更し，計算効率の比較を行う．扱う問題は 3 次元骨組構造物の弾性解析とし，様々な規模の解析モデルに対して全ポテンシャルエネルギー最小化問題を解き，アルゴリズムの違いによる収束性の違いを考察する．

2. 弾性解析と全ポテンシャルエネルギー最小化

外力ベクトルを $\boldsymbol{p}$，線形弾性剛性行列を $\boldsymbol{K}$ とし，節点変位ベクトル $\boldsymbol{x}$ を求めることを考える．式 (1), 式 (2) はそれぞれ剛性方程式，ポテンシャルエネルギー最小化の式を表す．

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{p} \tag{1}$$

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{K}\boldsymbol{x} - \boldsymbol{p}^\top \boldsymbol{x} \tag{2}$$

式 (1) を解くことと式 (2) を解くことは等価である．本報告では，式 (2) を様々なアルゴリズムで解く数値実験を行う．

3. 勾配法

本報告では，様々な規模の解析モデルに対して最急降下法，共役勾配法，加速勾配法の 3 つの勾配法を式 (2) に適用し，収束性能の比較を行う．

3.1. 最急降下法

最急降下法は，関数の勾配が最も急な方向に探索の方向をとりながら最小点にたどり着く方法である．単純で計算量が少ない方法だが，関数の非線形性が強い場合には最適解への収束性能が悪化しやすい．最急降下法のアルゴリズムを以下に示す．

Step 0　$k = 0$ とし，初期ベクトル $\boldsymbol{x}^{(k)}$, 初期探索ベクトル $\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$，収束判定値 ε を選ぶ．
Step 1　ステップ幅 $\alpha^{(k)}$ を定める．
Step 2　$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)}\boldsymbol{d}^{(k)}$ とする．
Step 3　$\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right\| \leq \varepsilon$ となれば解析終了．
Step 4　$\boldsymbol{d}^{(k+1)} = -\nabla f(\boldsymbol{x}^{(k+1)})$ とする．
Step 5　$k = k + 1$ として **Step 1** へ戻る．

収束判定の方法は複数あるが，本報告では最も簡便な方法として，設計変数の変化量 (前ステップとの設計変数ベクトルの差のノルム) が十分に微小となった段階で解析を終了とした．以降に説明する他のアルゴリズムでも同様の方法としている．

日本建築学会情報システム技術委員会
第42回情報・システム・利用・技術シンポジウム論文集，260–263，2019年12月，東京
Proceedings of the 42nd Symposium on Computer Technology of Information, Systems and Applications，AIJ，260–263，Dec.，2019，Tokyo

3.2. 共役勾配法

探索方向をお互いに共役方向に取ることで探索の効率化をはかる手法である. 共役勾配法も目的関数の勾配ベクトルを利用する非線形最適化問題の解法の 1 つである. 共役方向の取り方には複数の方法があるが，本研究ではフレッチャー・リーブスの式 [6] を用いた．共役勾配法のアルゴリズムを以下に示す.

Step 0 $k=0$ とし，初期ベクトル $\boldsymbol{x}^{(k)}$，初期探索ベクトル $\boldsymbol{d}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)})$，収束判定値 ε を選ぶ.

Step 1 ステップ幅 $\alpha^{(k)}$ を定める.

Step 2 $\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\alpha^{(k)}\boldsymbol{d}^{(k)}$ とする.

Step 3 $\left\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\right\|\leq\varepsilon$ となれば解析終了.

Step 4 $\boldsymbol{d}^{(k+1)}=-\nabla f(\boldsymbol{x}^{(k+1)})+\dfrac{\nabla f(\boldsymbol{x}^{(k+1)})\nabla f(\boldsymbol{x}^{(k+1)})}{\nabla f(\boldsymbol{x}^{(k)})\nabla f(\boldsymbol{x}^{(k)})}\boldsymbol{d}^{(k)}$ とする.

Step 5 $k=k+1$ として **Step 1** へ戻る.

3.3. 加速勾配法

加速勾配法は，最急降下法の欠点である解の収束速度を大幅に改善し，大規模問題を解く効果的な手法として注目を集めている手法である．Nesterov の提案した加速スキーム [7] が広く知られているが，この方法は探索方向が目的関数の降下方向とは限らず，目的関数値が常に減少するとは限らないという欠点がある．これに対し，探索方向が目的関数の増加方向に転じた際，加速の際に用いるパラメーターを初期値に戻すことにより，目的関数値を単調に減少させる方法が適応再スタート付き加速勾配法として O′Donoghue and Candès により提案されており [8]，本報告では同法のことを加速勾配法と位置付けている．加速勾配法のアルゴリズムを以下に示す.

Step 0 $k=0$ とし，初期ベクトル $\boldsymbol{x}^{(k)}$，初期探索ベクトル $\boldsymbol{d}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)})$，収束判定値 ε を選ぶ.
$\boldsymbol{y}^{(k)}=\boldsymbol{x}^{(k)}$，$\tau^{(k)}=1$ とおく.

Step 1 ステップ幅 $\alpha^{(k)}$ を定める.

Step 2 $\boldsymbol{x}^{(k+1)}=\boldsymbol{y}^{(k)}+\alpha^{(k)}\boldsymbol{d}^{(k)}$ とする.

Step 3 $\left\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\right\|\leq\varepsilon$ となれば解析終了.

Step 4 $\nabla f(\boldsymbol{y}^{(k)})^{\top}(\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)})\leq 0$ ならば
$\tau^{(k+1)}=\dfrac{1}{2}\left(1+\sqrt{1+\left(2\tau^{(k)}\right)^2}\right)$
$\boldsymbol{y}^{(k+1)}=\boldsymbol{x}^{(k)}+\dfrac{\tau^{(k)}-1}{\tau^{(k+1)}}(\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)})$
と更新し，そうでなければ
$\tau^{(k+1)}=1$，$\boldsymbol{y}^{(k+1)}=\boldsymbol{x}^{(k+1)}$
と更新する.

Step 5 $k=k+1$ として **Step 1** へ戻る.

4. ステップ幅の設定

勾配法の収束性能は，ステップ幅の設定方法に大きく依存する．ステップ幅の設定は, 各計算ステップごとにラインサーチを行い逐次設定する方法と，最適化計算の全ステップで同じ値を用いる方法とがある．本報告では，前者として黄金分割法及び armijo 基準によるバックトラック法と，後者として Gershgorin の定理を応用した方法を採用し，各手法の収束性能の比較を行う.

4.1. 黄金分割法

黄金分割法は古典的なラインサーチの方法のひとつで，極値が存在するとわかっている範囲を逐次的に狭めていき探索方向に目的関数を最も減少させるステップ幅を厳密に求める手法である．黄金分割法のアルゴリズムを以下に示す.

Step 0 k ステップ目における設計変数ベクトル $\boldsymbol{x}^{(k)}$，探索ベクトル $\boldsymbol{d}^{(k)}$ を既知量とする．$i=0$，$\eta=\dfrac{1+\sqrt{5}}{2}$，とし，初期探索範囲 $l^{(i)}$〜$u^{(i)}$ 及び収束判定値 ε を選ぶ.

Step 1 $u^{(i)}-l^{(i)}\leq\varepsilon$ となれば解析終了．$\alpha^{(k)}=\dfrac{u^{(i)}-l^{(i)}}{2}$ とする.

Step 2 $\alpha_1^{(i)}=\dfrac{l^{(i)}\cdot\eta+u^{(i)}}{1+\eta}$，$\alpha_2^{(i)}=\dfrac{l^{(i)}+u^{(i)}\cdot\eta}{1+\eta}$ とする.

Step 3 $f(\boldsymbol{x}^{(k)}+\alpha_1^{(i)}\cdot\boldsymbol{d}^{(k)})\geq f(\boldsymbol{x}^{(k)}+\alpha_2^{(i)}\cdot\boldsymbol{d}^{(k)})$ ならば
$l^{(i+1)}=\alpha_1^{(i)}$，$u^{(i+1)}=u^{(i)}$ と更新し，そうでなければ
$l^{(i+1)}=l^{(i)}$，$u^{(i+1)}=\alpha_2^{(i)}$ と更新する.

Step 4 $i=i+1$ として **Step 1** へ戻る.

4.2. armijo 基準によるバックトラック法

armijo 基準によるバックトラック法はラインサーチの方法のひとつで，armijo 基準を満足する範囲でステップ幅をなるべく大きく取り，探索方向に目的関数を最も減少させるステップ幅を近似的に求める手法である．armijo 基準によるバックトラック法のアルゴリズムを以下に示す.

Step 0 k ステップ目における設計変数ベクトル $\boldsymbol{x}^{(k)}$, 探索ベクトル $\boldsymbol{d}^{(k)}$ を既知量とする．$i=0$, armijo 基準のパラメータを $a\in(0,1)$, 縮小率を $b\in(0,1)$, 初期ステップ幅を $e^{(i)}$ とする.

Step 1 armijo 基準
$f(\boldsymbol{x}^{(k)}+e^{(i)})-f(\boldsymbol{x}^{(k)})\leq a\cdot e^{(i)}\cdot\nabla f(\boldsymbol{x}^{(k)})\cdot\boldsymbol{d}^{(k)}$
を満たせば解析終了．$\alpha^{(k)}=e^{(i)}$ とする.

Step 2 $e^{(i+1)}=b\cdot e^{(i)}$ とする.

Step 3　$i = i + 1$ として **Step 1** へ戻る．

4.3. Gershgorin の定理

厳密にラインサーチを行うと，最適化の総解析ステップ数を減少させることはできても，ラインサーチ自体に計算コストがかかってしまい, アルゴリズム全体としては計算効率が悪化する可能性がある. そこで本研究では，ラインサーチを行わずに剛性行列の固有値の情報を用いてステップ幅を定めることを考える．$\nabla f(\boldsymbol{x})$ の Lipschitz 定数を L とするとき，$f(\boldsymbol{x})$ が Lipschitz 連続かつ凸関数であれば, $\alpha^{(k)} \in (0, 1/L])$ を満たすように選ぶと解の収束が保証される．全体剛性行列を $\boldsymbol{K}$ としたとき，$f(\boldsymbol{x})$ が凸 2 次関数の場合は，L は $\boldsymbol{K}$ の最大固有値 $\lambda_{\max}$ に等しい．ただし, 解析規模が大きくなると固有値解析に大幅な計算時間を有する．$\boldsymbol{K}$ の固有値が全て非負であることを考慮すれば, Gershgorin の定理より, L の上界は $L_{\max} = \max\left\{K_{ii} + \sum_{j \neq i} |K_{ij}| \,\middle|\, i = 1, \cdots, D\right\}$ で求められ，こちらは固有値解析よりも高速に計算することができる (ここで，D は $\boldsymbol{K}$ のサイズ)．よってステップ幅は $\alpha^{(k)} = \dfrac{1}{L_{max}}$ で定めることで，計算の効率化を図る．線形弾性解析の場合には，剛性行列は定数となるので，ステップ幅 $\alpha^{(k)}$ はアルゴリズム全体を通じて一定値となる．

5. 解析モデル

図に示すようなスパン 20m，ライズ 6m のラチスシェル (裁断球殻) を考える．部材は $N \times N$ のグリッド状に配置され，部材断面は全て 10cm 角の正方形断面とする (図の例では $N = **$)．材料は鉄骨を想定し，ヤング係数を 205GPa，単位体積重量を 77kN/m^3 とする．境界条件は 4 隅をピン支持とし，外力は自重のみとする．最急降下法 (SDM), 加速勾配法 (AGM), 共役勾配法 (CG) の 3 つの勾配法を用い，ステップ幅については Gershgorin の定理 (gershgorin), 黄金分割法 (golden),armijo 基準によるバックトラック法 (armijo) の 3 パターンについて比較検討を行い，勾配法 3 種類×ステップ幅決定方法　3 種類の合計 9 種類のアルゴリズムを用いて全ポテンシャルエネルギー最小化問題 (2) を解く．スパンの分割数 N の値を小さな値から大きな値まで変化させ，問題の規模の違いによる収束性能の比較を行う．

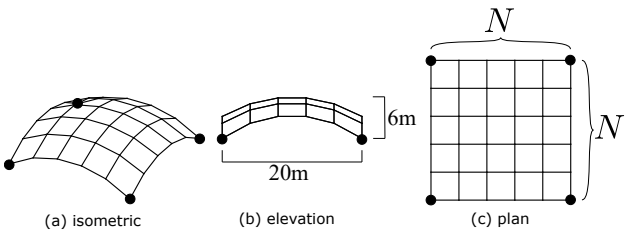


図 1　Analytical model (ex. $N = 5$)

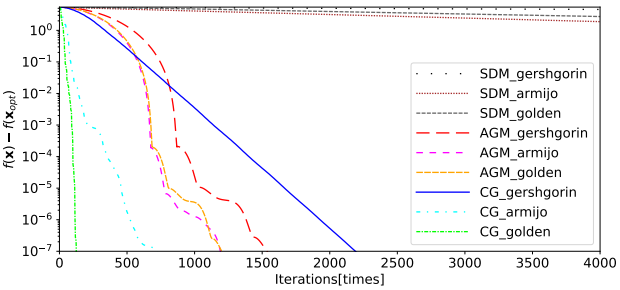
なお，プログラムは Python により記述し，計算効率の条件を揃えるために，ベクトル演算などは numpy などのライブラリを用いず，すべてループ処理により計算する．重量言語と異なり計算時間は大きくなるが，本報告の狙いは各アルゴリズムの性能比較を行うことにある．

最適化に際し，収束判定定数はすべてのアルゴリズムにおいて $\varepsilon = 1.0^{-9}$ とし，黄金分割法における初期探索幅は $l^{(0)} = 0$，$u^{(0)} = 1$ とし，armijo 基準のパラメータは $a = 0.5$，縮小率は $b = 0.8$ とする．また，初期解は一律に $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ とする．

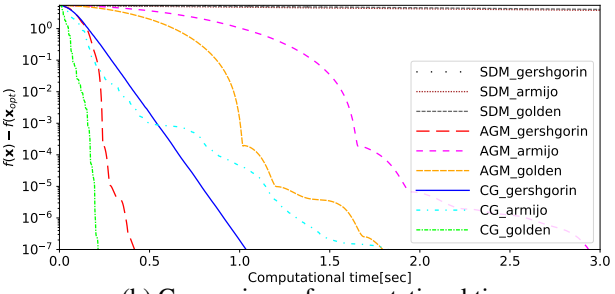
6. 数値解析結果

スパンの分割数を $N = 5, 10, 15$ とした場合の数値解析結果を図 2〜4 にそれぞれ示す．各図において，縦軸には目的関数の最適解との差を対数軸で取り，横軸には (a) 解析ステップ数，(b) 計算時間をそれぞれ取り，目的関数である全ポテンシャルエネルギーの収束履歴を示している．なお，ここでの解析ステップ数とは，最適化計算における k の値であり，ラインサーチを行う際の内部反復回数は含まれてないことに注意する．

SDM はステップ幅の決定方法に関わらず，CG 及び AGM と比べて圧倒的に収束性能に劣る結果となったため，議論の対象から除外し，CG と AGM について比較する．各図を見ると，$N = 5, 10, 15$ のすべての場合において，CG_golden(ステップ幅を黄金分割法で決め，共役勾配法で解く場合) が最も解析ステップ数が少なくなる結果となった．しかしながら，計算時間に着目すると，$N = 5$ の場合では CG_golden 最も計算時間が短いが，$N = 10$ になるとわずかに AGM_gershgorin(gershgorin の定理を応用してステップ幅一定として加速勾配法で解く場合) の方が計算時間が短くなり，$N = 15$ のときにはその差は拡大した．アルゴリズムごとに見ていくと，CG では，golden(黄金分割法) が最も計算効率が高く，解析ステップ数，計算時間ともに最小となった．それに対して armijo(armijo 基準によるバックトラック法) は計算効率が最も悪く，多くの計算時間を要した．ただし，a，b のパラメータ次第では改善される可能性はある．gershgorin(Gershgorin の定理を応用した固定ステップ幅) では，解析ステップ数は最も多いが，ラインサーチの反復計算がない分だけ計算時間は armijo より短く，N の値が大きくなるにしたがって golden との差も小さ

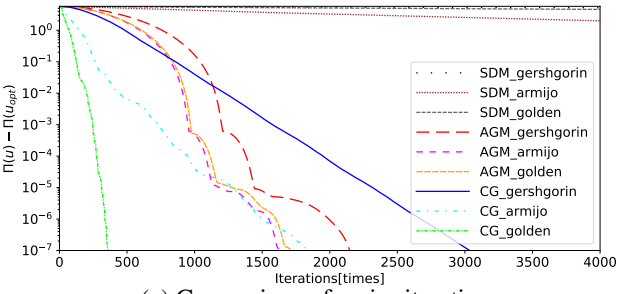


(a) Comparison of major iterations

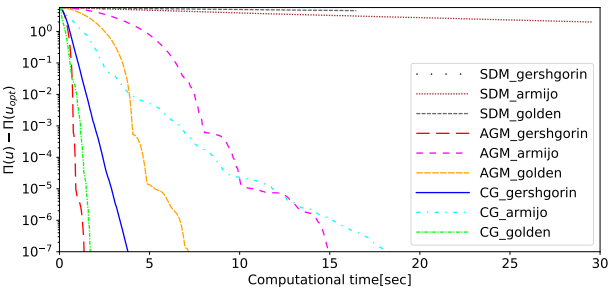


(b) Comparison of computational time

図 2　Convergence history of objective function ($N = 5$)



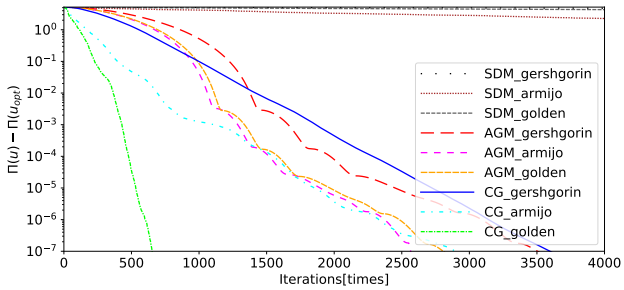
(a) Comparison of major iterations



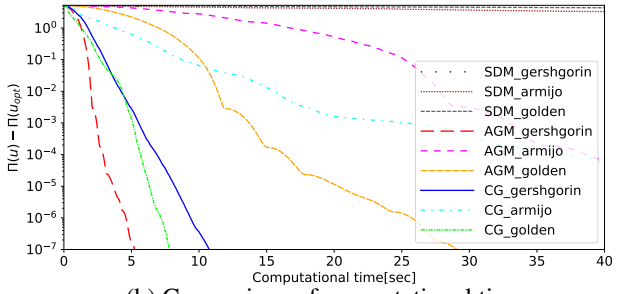
(b) Comparison of computational time

図 3　Convergence history of objective function ($N = 10$)

くなった．一方で AGM では，gershgorin が最も計算効率が良く，その傾向は N の値が大きいほど顕著に表れ，$N = 15$ のときではすべてのアルゴリズムの中で最小の計算時間となった．AGM の場合でも CG と同様，今回のパラメータでは armijo との相性は悪く，gershgorin や golden と比較して多くの計算時間を要した．



(a) Comparison of major iterations



(b) Comparison of computational time

図 4　Convergence history of objective function ($N = 15$)

7. まとめ

本報告では，3 次元骨組構造物の弾性解析と等価な問題である全ポテンシャルエネルギー最小化問題を SDM，CG，AGM の 3 つの勾配法を用いて解き，ステップ幅の決定方法や問題の規模による収束性能の違いを比較する数値実験を行った．規模の小さい問題に対しては，CG_golden が最も計算効率が高いが，解析規模が大きくなると，AGM_gershgorin が計算効率の点において優位性を示すことが確認された．

[参考文献]

1）T. Suzuki. *Stochastic optimization*. Kodansha, 2015. (in Japanese).

2）Y. Kanno. Accelerated proximal gradient method for elastplastic analysis of large-scale trusses. *Proceedings of OPTIS 2016 (Japan Society of Mechanical Engineers)*, No. 1213, 2016.1. (in Japanese).

3）S. Fujita, Y. Kanno, and M. Ohsaki. Accelerated gradient method for geometrically nonlinear analysis of framed structures. *Proceedings of OPTIS 2016 (Japan Society of Mechanical Engineers)*, No. 1212, 2016.1. (in Japanese).

4）Y. Kanno. Accelerated proximal gradient method for equilibrium analysis of elastoplastic spatial truss structures. *IASS Annual Symposium—Spatial Structures in the 21st Century*, 2016.9.

5）S. Fujita and Y. Kanno. Application of accelerated gradient method to equilibrium analysis of trusses with nonlinear elastic materials. *J. Struct. Constr. Eng., AIJ*, Vol. 84, No. 763, pp. 1223–1230, 2019.9. (in Japanese).

6）R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, Vol. 7, No. 2, pp. 149–154, 1964.1.

7）Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Doklady*, No. 269, pp. 372–376, 1983.

8）B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, Vol. 15, pp. 715–732, 2015.7.