

2023年度未踏ターゲット事業（量子コンピューティング技術を活用したソフトウェア開発分野）

量子コンピューターを用いた境界要素法の開発

－ 計算力学への応用を目指して －

齋藤隆泰（群馬大学大学院理工学府 環境創生部門）

□ 背景・目的

近年、量子コンピューターに注目が集まっている。令和4年に内閣府が掲げた量子未来社会ビジョンでは、量子コンピューターを活用すべき分野の1つとしてHPC(High Performance Computing)との連携を挙げている。そのため、量子コンピューターを用いて工学の様々な問題を解決するためのシミュレーション手法を開発することは重要である。事実、シミュレーションを援用して工学の諸問題を解決する「**計算力学**」分野においても、年々、その注目が集まっている。

➡ **量子コンピューターを用いた境界要素法の開発**を
目的に本プロジェクトを実施



図1：Science Directで「量子コンピューター」「シミュレーション」で検索ヒットした論文数

□ 開発したソフトウェアの特徴

(新規性・優位性)

- 2次元ラプラス方程式, Helmholtz方程式を対象
- 有限要素法を対象とした例[文献1]は、近年見受けられるものの境界要素法を対象とした例はない(未踏性が高い)
- 無限遠を含む場合や、波動問題に特に有効
- 将来的に、超大規模解析が実行可能

□ 解決する課題と社会への影響

- これまで解けなかったような超大規模問題を解ける可能性がある
- 境界要素法に限らず、有限要素法等を含め、大規模な代数方程式を解く問題へも拡張可能
- 工学の様々な未踏シミュレーションの可能性

[文献1]A. Mielke and T. Ricken: Finite element analysis of a 2D cantilever on a noisy intermediate-scale quantum computer, Proceedings in Applied Mathematics & Mechanics, (2021), DOI:10.1002/pamm.202100246

■ 解析のポイントと通常境界要素法の解析手順(有限要素法も同様)

- ① 形状データの入力 (どの数値計算でも避けて通れない) ①②⑤は有限要素法でも同様
- ② 代数方程式における係数行列 A , B の作成 (数値積分を伴う) ①⑤はどの手法でも必ず必要
- ③ 係数行列 B とベクトル c による行列ベクトル積 Bc の計算 ここでは④の代数方程式の解法に
- ④ 線形代数方程式の計算 ($Ax=b$) 古典・量子ハイブリッド解法を提案
- ⑤ 結果の出力 (どの数値計算でも避けて通れない) (境界要素法のボトルネック解消)

■ 解析例(2次元ラプラス方程式の場合) とアピールポイント

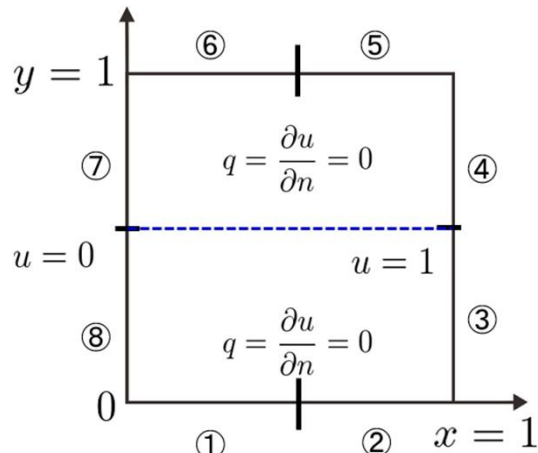
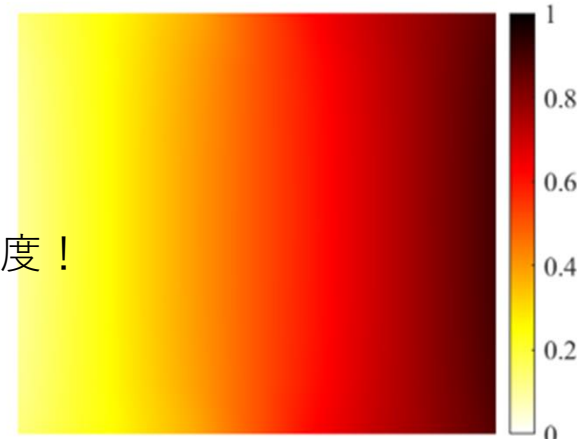


図2: 解析モデル (定常熱伝導問題)

classical	quantum	相対誤差%
0.241415 (u)	0.243607	0.908161
0.758588(u)	0.762996	0.58113
1.05863(q)	1.05343	0.491142
1.05863(q)	1.053	0.491142
0.758588(u)	0.762996	0.58113
0.241415(u)	0.243607	0.908161
-1.05859(q)	-1.04956	0.853568
-1.05859(q)	-1.04956	0.853568

表1: 境界要素①-⑧における古典計算, 古典・量子ハイブリッド計算結果の比較



高精度!

図3: 表1の結果を使った古典計算による解析モデル内の定常温度場.

HHL[文献2]やLCU[文献3]を用いた境界要素法を実行できた

■ 今後の課題

- 上記②③の量子アルゴリズムによる計算, 実機での計算, 量子コンピューター用の境界要素アルゴリズムの検討

[文献2] W. Harrow, A. Hassidim, and S. Lloyd: Quantum algorithm for linear systems of equations, *Physical Review Letters*, **103**, 150502, (2009)

[文献3] A. M. Childs, R. Kothari and R. D. Somma: Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM Journal on Computing*, **46**(6), (2017)