Parallelized Conjugate Gradient Method for 2D High-order Spectral Element Galerkin Method using Global Communication

Nicholas Troescher – Higdon Group – Department of Chemical & Biomolecular Engineering

Introduction

A parallelized (distributed memory) method for solving Poisson's equation using a High-order spectral element Galerkin method (GM) with an arbitrarily preconditioned conjugate gradient iterative method (PCGM) is presented in this work. A typical PCGM requires at least two global reductions among elements to determine the search direction, namely $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Additionally, elemental domains $\boldsymbol{\Omega}$ must exchange border residuals on a subdomain $\boldsymbol{\Gamma}$ with all neighboring elements. This method avoids performing a local exchange by inserting nodal residuals that need to be exchanged ($\mathbf{y}_\Gamma$) into the all-reduce communication required to calculate $\boldsymbol{\alpha}$. In application, this method provides a faster alternative for a latency bound locally communicated GM solver.

Linear System Formulation

Weighted Residual Method

$$\int_V \phi_\alpha \left(\nabla^2 u \;+ f \right)\, dV = 0$$

Isoparametric Mapping

$$\partial_x \;=\; \xi_x\, \partial_\xi \qquad \partial_y \;=\; \eta_y\, \partial_\eta$$

$$J_{xy} = \; x_\xi\, y_\eta \;-\; y_\xi\, x_\eta$$

Interpolate with Lagrange Polynomials *h*

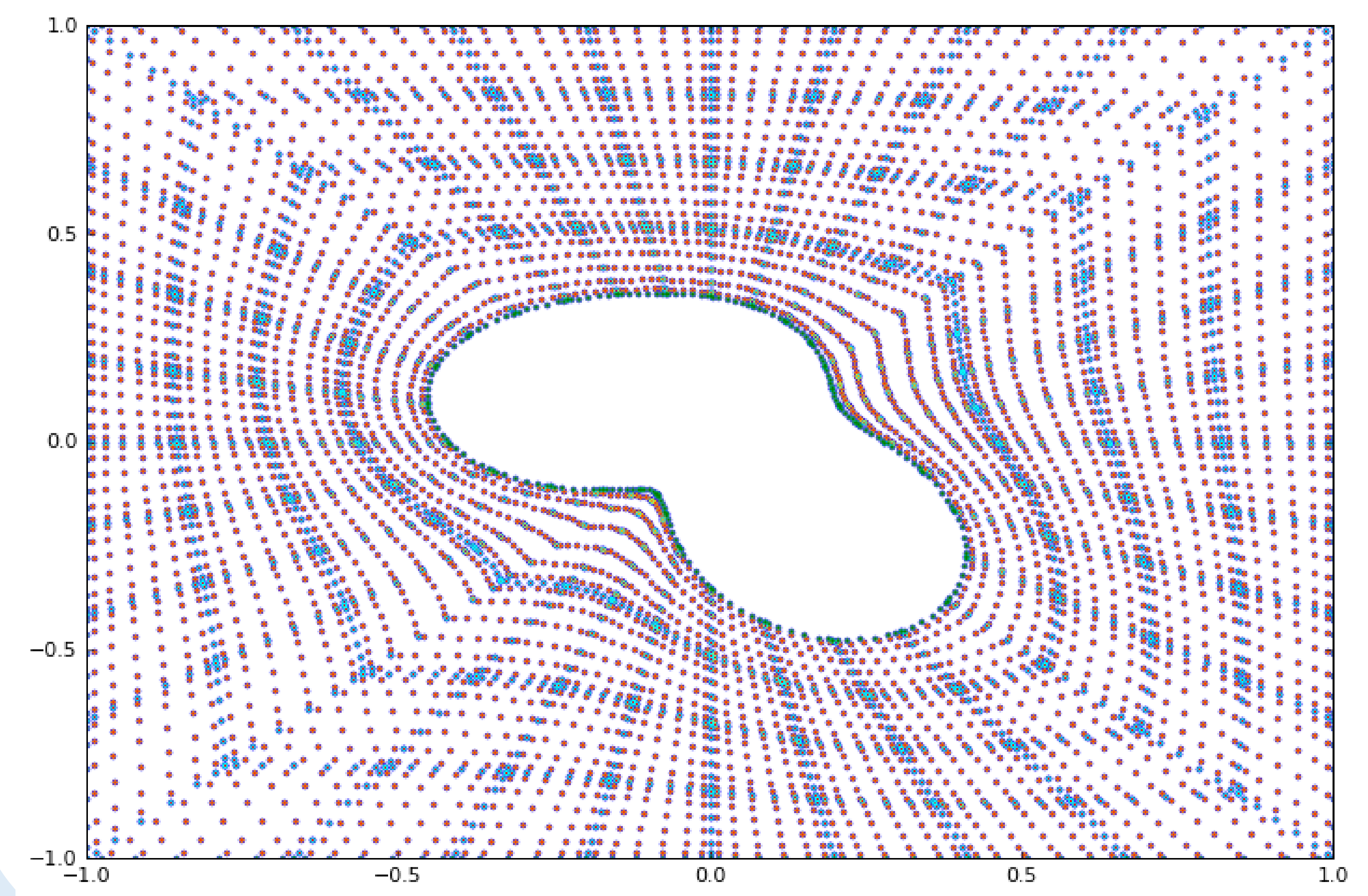
$$u = \sum_{m,n} h_m(\xi)\, h_n(\eta) u_{mn}$$

$$\phi_\alpha = h_p\,(\xi)\, h_q(\eta)$$

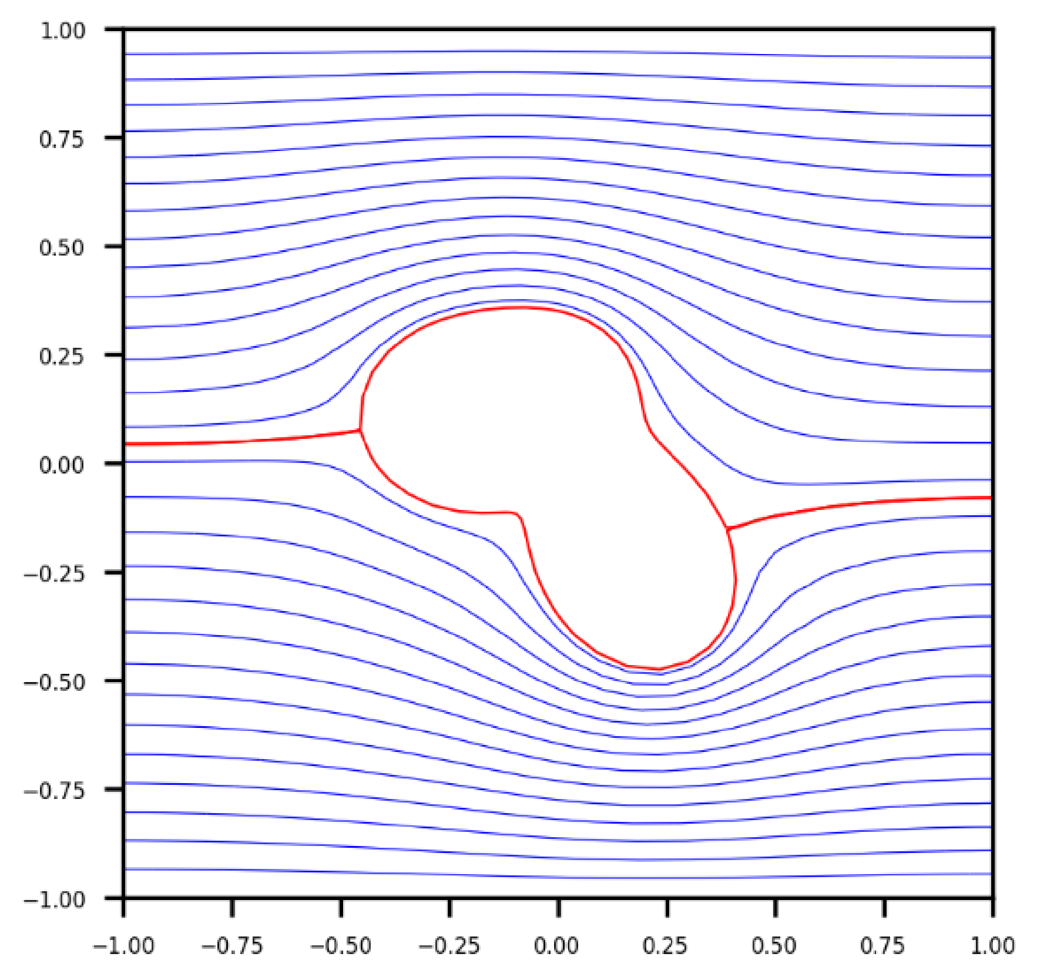
$$h_\beta(x) = \sum_{\substack{m=0\\ m\neq\beta}}^{n} \frac{x - x_m}{x_\beta - x_m}$$

$$\boxed{\begin{aligned} &\sum\nolimits_{m,n} \left(C_{pqmn}\;\, u_{mn}\right) \;+\; J_{xy} w_p w_q\, f_{pq} = 0 \\ &C_{pqmn} = \sum\nolimits_{i,j} J_{xy}\, {\xi_x}^2 \left(w_i w_j\, D_{ip}\, \delta_{jq}\, D_{im}\, \delta_{jn}\right) + \; J_{xy}\, \eta_y^2\, \left(w_i w_j\, \delta_{ip}\, D_{jq} \delta_{im}\, D_{jn}\right) \end{aligned}}$$

Elemental Mapping ($\boldsymbol{n_R \times n_\theta}$)



Solution (Flow Past an Arbitrary Object)



Method

Split Preconditioned Conjugate Gradient

$$\mathbf{C\,u = f}$$

$$\mathbf{M \sim C} \qquad \mathbf{M = LL^T}$$

$$\boxed{\mathbf{L^{-1}CL^{-T}\,x = L^{-1}f} \qquad \mathbf{u = L^{-T}x}}$$

Parallel Implementation

(1) $Each\ processor\ given\ n_R n_\theta /\, p\ elements$

(2) $i^{th}\ processor\ constructs\!:\ \mathbf{C}^{(i)}, \mathbf{u}_o^{(i)}, \mathbf{L}^{(i)}\ and\ \mathbf{f}^{(i)}$

(3) $\mathbf{r}_o'^{(i)} = \mathbf{f}^{(i)} - \mathbf{C}^{(i)}\, \mathbf{u}_o^{(i)}$

(4) $\mathbf{r}_o^{(i)} = \mathbf{L}_{(i)}^{-1}\, \mathbf{r}_o'^{(i)}$

(5) $\mathbf{p}_o^{(i)} = \mathbf{L}_{(i)}^{-\mathrm{T}}\, \mathbf{r}_o^{(i)}$

Algorithm Characteristics

- Two global reductions required for α and β, regardless of parallelism
- Arbitrary agglomeration can be used as there are no local communications
- Algorithm effectively parallelizes all BLAS 1,2 operations

Algorithm

(let I and Γ denote interior and border nodes, respectively.)

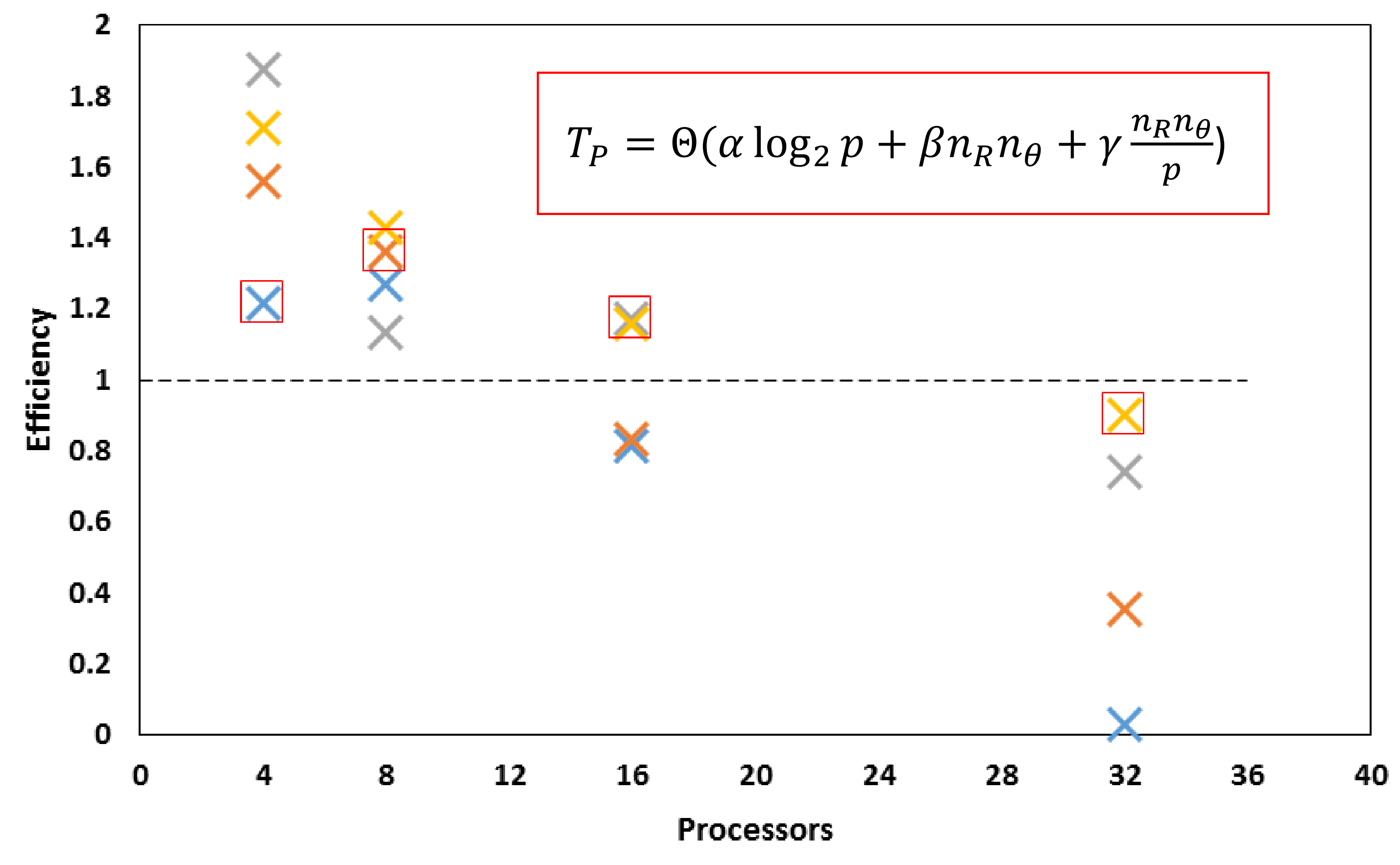
```
$For\ j = 0,..1, until\ convergence\ Do$:
    $\mathbf{y}^{(i)} = \mathbf{C}^{(i)}\mathbf{p}_j^{(i)}$
    $\alpha_j' = \left(\mathbf{r}_j^{(i)}, \mathbf{r}_j^{(i)}\right)$
    $\alpha_j'' = \left(\mathbf{y}^{(i)}, \mathbf{p}_j^{(i)}\right)$
    $Partition\left[\mathbf{y}^{(i)}\right] \rightarrow \mathbf{y}_I^{(i)}, \mathbf{y}_\Gamma^{(i)}$
    $Sparsely\ fill\ \mathbf{y}_\Gamma\ with\ \mathbf{y}_\Gamma^{(i)}$
    $comm.Allreduce[\alpha_j', \alpha_j'', \mathbf{y}_\Gamma]$
    $Extract\ \mathbf{y}_\Gamma^{(i)}\ from\ \mathbf{y}_\Gamma$
    $Combine[\mathbf{y}_\Gamma^{(i)}, \mathbf{y}_I^{(i)}] \rightarrow \mathbf{y}^{(i)}$
    $\alpha_j = \alpha_j' \,/\, \alpha_j''$
    $\mathbf{u}_{j+1}^{(i)} = \mathbf{u}_j^{(i)} + \alpha_j \mathbf{p}_j^{(i)}$
    $\mathbf{r}_{j+1}^{(i)} = \mathbf{r}_j^{(i)} - \alpha_j \mathbf{L}_{(i)}^{-1}\mathbf{y}^{(i)}$
    $\beta_j' = \left(\mathbf{r}_{j+1}^{(i)}, \mathbf{r}_{j+1}^{(i)}\right)$
    $comm.Allreduce[\beta_j']$
    $\beta_j = \beta_j' \,/\, \alpha_j'$
    $\mathbf{p}_{j+1}^{(i)} = \mathbf{L}_{(i)}^{-\mathrm{T}}\mathbf{r}_{j+1}^{(i)} + \beta_j \mathbf{p}_j^{(i)}$
$EndDo$
```

Conclusion

Ultimately, this algorithm details a more efficient choice when a computation is bound by latency using a PCGM on a GM solver. As long as $\boldsymbol{\alpha \ggg \beta n_R n_\theta}$, meaning a local residual exchange would cost more than the additional bandwidth for the global residual exchange, one should expect favorable scaling.

Note that this study considers an arbitrary preconditioner (all simulations ran for 100 iterations regardless of error), as number of iterations plays an obvious role in timing iterative methods. High-order spectral elements methods are notoriously poorly conditioned, so a separate study entirely would need to be done to characterize a parallel algorithm that considers an optimal preconditioner.

Scaling Analysis



$$T_P = \Theta\left(\alpha \log_2 p + \beta n_R n_\theta + \gamma \frac{n_R n_\theta}{p}\right)$$

Strong and Weak Scaling Analysis: "×" Blue, Orange, Grey, and Yellow correspond to data sizes with $(n_R = 8, n_\theta = 32)$, $(n_R = 8, n_\theta = 64)$, $(n_R = 16, n_\theta = 64)$, and $(n_R = 16, n_\theta = 128)$, respectively. "■" markers denote series for weak scaling analysis. All runs used 64 basis points per element, and 100 iterations.

References

1. Taneja A., & Higdon J. J. (2018) A fully-coupled discontinuous Galerkin spectral element method for two-phase flow in petroleum reservoirs. *Journal of Comp Phys, 352* pp. 341-372

2. Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems.* Philadelphia, PA: Society of Industrial and Applied Mathematics