

# Tomografia de Resistividade Elétrica em Ambiente HPC: Revisão de um Estudo Recente

Ana Clara Lannes<sup>1</sup>, Alfredo Goldman<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística – Universidade de São Paulo (USP)  
Centro de Competência em Software Livre

anaclaralannes@ime.usp.br, gold@ime.usp.br

**Abstract.** *Electrical Resistivity Tomography (ERT) is a geophysical technique applied to subsurface investigation, with applications in mineral exploration, environmental monitoring, and large-scale geological studies. Efficient data interpretation demands advanced algorithms capable of processing large volumes of information. Due to the complexity and size of these datasets, the use of High-Performance Computing (HPC) systems becomes essential to enable intensive simulations. This work reviews a recent study that proposes a massively parallel implementation for the modeling and inversion of 3D ERT data using the open-source PFLOTRAN code, with an emphasis on computational performance aspects observed in HPC environments.*

**Resumo.** *A tomografia de resistividade elétrica (ERT) é uma técnica geofísica aplicada à investigação do subsolo, com usos em exploração mineral, monitoramento ambiental e estudos geológicos em larga escala. A interpretação eficaz dos dados exige algoritmos avançados capazes de lidar com grandes volumes de informação. Dada a complexidade e dimensão desses dados, o uso de sistemas de computação de alto desempenho (HPC) torna-se essencial para viabilizar simulações intensivas. Este trabalho revisa um estudo recente que propõe uma implementação massivamente paralela para modelagem e inversão de dados ERT em 3D, utilizando o código aberto PFLOTRAN, com foco no desempenho computacional observado em ambientes HPC.*

## 1. Introdução

A Tomografia de Resistividade Elétrica (ERT) com corrente contínua é amplamente empregada em aplicações ambientais, minerais e geotécnicas devido à sua capacidade de revelar estruturas subsuperficiais com alta sensibilidade [Jaysaval et al. 2022]. Com o avanço de sistemas multicenais, tornou-se comum a aquisição de grandes volumes de dados, exigindo algoritmos de modelagem e inversão que sejam não apenas precisos, mas também altamente escaláveis [Wright 2006]. A inversão, em particular, transforma dados brutos em imagens tridimensionais da condutividade elétrica do subsolo.

## 2. Metodologia Computacional para ERT em HPC

### 2.1. Estrutura Algorítmica e Estratégia de Particionamento

A implementação analisada baseia-se na discretização por volumes finitos (FV) da equação de Poisson, responsável por modelar o potencial elétrico no meio tridimensional [Jahandari and Farquharson 2014]. Este método opera sobre a forma integral da

equação governante e foi escolhido por sua capacidade de lidar eficientemente com malhas estruturadas em geometrias complexas, mantendo simplicidade computacional. A solução do sistema linear resultante é realizada por métodos iterativos baseados na biblioteca PETSc, com uso de solvers como GMRES e pré-condicionadores configuráveis [Jaysaval et al. 2022]. A inversão dos dados ERT é tratada como um problema de otimização não linear, resolvido via método de Gauss–Newton regularizado, com cálculo explícito do Jacobiano usando o método de estados adjuntos [Wright 2006].

Quanto à paralelização, o PFLOTRAN adota decomposição de domínio, com distribuição espacial da malha computacional entre os processos MPI. Em malhas estruturadas, como as utilizadas neste trabalho, o particionamento ocorre de forma cartesiana, preferencialmente balanceando a quantidade de células por processo e minimizando a superfície de contato entre subdomínios, o que reduz o volume de comunicação entre nós. Cada subdomínio gerenciado por um processo MPI armazena apenas os dados locais e uma camada de células fantasmas para troca de informações nas fronteiras. O uso de estruturas de dados distribuídas da PETSc facilita essa comunicação de forma eficiente [Jaysaval et al. 2022] [Balay et al. 2021]. Para malhas não estruturadas (não utilizadas neste estudo, mas implementadas no PFLOTRAN), o particionamento pode ser feito por meio do ParMETIS, que visa minimizar o custo de comunicação interprocessos [Lichtner et al. 2015].

Essa abordagem permite ao PFLOTRAN atingir excelente escalabilidade até o limite prático de cerca de 10.000 DOFs por processo, conforme validado por [Jaysaval et al. 2022]. O projeto modular do código também favorece extensões para estratégias híbridas (MPI + OpenMP), úteis para exploração de arquiteturas multicore contemporâneas.

## 2.2. Ambiente de Execução e Configuração Experimental

O ambiente de execução foi baseado no sistema operacional Linux (Xubuntu 20.04 LTS), com o uso dos compiladores GCC 9.4.0 e GFortran. A instalação do PFLOTRAN envolveu a configuração e compilação da biblioteca PETSc (versão 3.16.6) por meio do script de configuração do próprio PFLOTRAN. As opções utilizadas incluíram:

```
./configure --download-fblaslapack --with-debugging=0
--COPTFLAGS=-O3 --FOPTFLAGS=-O3 make -f GNUmakefile
process_model=ert mpirun -np 32 ./pflotran -pflotranin input.in
```

A execução paralela foi realizada com o OpenMPI 4.1.1. Durante os testes, o número de núcleos foi variado entre 10, 20 e 64, conforme especificado na Tabela 1. Os arquivos de entrada utilizados incluíram definição da malha tridimensional estruturada, blocos de propriedades elétricas, posições dos eletrodos, corrente injetada e parâmetros de inversão.

### 3. Análise Experimental de Desempenho

Tabela 1. Testes realizados na máquina 4x AMD Opteron 6376 (16c) 2.3 GHz

Tamanho da Matriz Teste	Quantidade de Núcleos Utilizados (de 64)	Memória Utilizada (de 256G/RAM)	Tempo total de Execução
100 × 80 × 120	20	54G	13.44 s
100 × 80 × 120	64	200G	1 min 09 s
120 × 96 × 144	10	70G	46.64 s
200 × 160 × 240	10	216G	2 min 22 s

#### 3.1. Escalabilidade e Uso de Memória

Durante os testes com o PFLOTTRAN, observou-se que o aumento no número de processos MPI utilizados levou a um crescimento considerável no consumo total de memória, mesmo quando mantidas constantes as dimensões da malha e os parâmetros de simulação. Esse comportamento decorre da arquitetura de paralelismo distribuído baseada em decomposição de domínio. Cada processo MPI gerencia localmente não apenas os dados do seu subdomínio, mas também estruturas auxiliares, como regiões fantasmas, buffers de fronteira, vetores de estado e informações de vizinhança. Essas estruturas tornam-se redundantes com o aumento no número de processos, elevando o consumo de memória.

Os resultados revelaram ainda que a eficiência paralela está fortemente condicionada à razão entre o número de graus de liberdade (DOFs) e o número de processos MPI. Configurações com baixa razão DOFs/processo, como no caso de 64 núcleos para a malha  $100 \times 80 \times 120$ , resultaram em degradação de desempenho. Esse comportamento é compatível com os limites de escalabilidade observados por [Jaysaval et al. 2022], que recomendam no mínimo 10.000 DOFs por processo para evitar ineficiências por sobrecarga de comunicação.

#### 3.2. Precisão Numérica e Qualidade dos Resultados

Todas as simulações utilizaram ponto flutuante em precisão dupla (64 bits), constante independentemente do número de processos. Assim, o crescimento da memória observada está relacionado à infraestrutura paralela, e não a mudanças na fidelidade numérica. A qualidade dos resultados permaneceu estável em todas as configurações testadas, o que foi validado também por [Jaysaval et al. 2022], que relataram erros médios relativos inferiores a 1% mesmo em execuções com até 131.072 processos MPI.

### 4. Considerações sobre Paralelismo

#### 4.1. Limitações de Escalabilidade e Sobreposição de Comunicação

A redução de tempo com o uso de múltiplos núcleos só é vantajosa até certo ponto. Para malhas menores, a sobrecarga de comunicação introduzida pode superar os benefícios da divisão de carga. Isso foi evidenciado nos testes com malha  $100 \times 80 \times 120$ , em que 20 núcleos apresentaram desempenho superior ao uso de 64 núcleos. Essa limitação reforça a necessidade de análise criteriosa entre granularidade computacional e paralelismo aplicado.

## 4.2. Potencial do Paralelismo Híbrido e Regionalização de Comunicação

O PFLOTRAN é compatível com paralelismo híbrido (MPI + OpenMP), em que múltiplos *threads* OpenMP compartilham o mesmo processo MPI. Essa abordagem pode reduzir o número de mensagens MPI e, conseqüentemente, a sobrecarga de comunicação interprocesso. Em arquiteturas com múltiplos núcleos por nó, essa estratégia permite regionalizar as comunicações, mantendo-as dentro do mesmo nó físico e otimizando o uso de memória compartilhada. Segundo [Hammond et al. 2014], tal abordagem pode melhorar significativamente a eficiência em sistemas HPC modernos. Embora ainda não tenha sido explorada neste estudo, essa funcionalidade representa uma direção promissora para investigações futuras.

## 5. Considerações Finais

A análise realizada permitiu não apenas confirmar os resultados do artigo de [Jaysaval et al. 2022], mas também revelar limitações práticas de portabilidade e eficiência computacional em ambientes não ideais. As medições demonstram a importância do balanceamento entre o tamanho do problema, a razão DOFs/processo e o grau de paralelismo.

Esta pesquisa conta com financiamento da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), por meio do Projeto Temático de número 19/26702-8.

## 6. Trabalhos Futuros

Pretende-se explorar o paralelismo híbrido e alternativas de implementação, como otimização de estruturas de dados, uso de vetorização, programação assíncrona ou aceleração por GPU. Também se planeja avaliar modelos matemáticos que ofereçam melhor estabilidade numérica e menor custo computacional.

## Referências

- Balay, S., Abhyankar, S., Adams, M., Benson, S., Brown, J., Brune, P., et al. (2021). *Petsc/tao users manual*, argonne national laboratory. Technical report, ANL-21/39-Revision 3.17, 2022. Available from: <https://petsc.org>.
- Hammond, G. E., Lichtner, P. C., and Mills, R. (2014). Evaluating the performance of parallel subsurface simulators: An illustrative example with pflotran. *Water resources research*, 50(1):208–228.
- Jahandari, H. and Farquharson, C. G. (2014). A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids. *Geophysics*, 79(6):E287–E302.
- Jaysaval, P., Hammond, G. E., and Johnson, T. C. (2022). Massively parallel modeling and inversion of electrical resistivity tomography data using pflotran. *Geoscientific Model Development Discussions*, 2022:1–26.
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., Mills, R., and Kumar, J. (2015). *Pflotran user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes*. Technical report.
- Wright, S. J. (2006). *Numerical optimization*.