

Performance do Método de Eliminação de Gauss para Resolver Sistema de Equações de Problemas de Fluxo em Meio Poroso Incompressíveis*

Tiago C. A. Amorim²

^aPetrobras, Av. Henrique Valadares, 28, Rio de Janeiro, 20231-030, RJ, Brasil

Abstract

Uma simulação de fluxo em meio poroso depende da resolução de um conjunto de equações não lineares a cada passo de tempo. Para modelos incompressíveis este conjunto de equações é *fracamente* não linear, e o Método do Ponto Fixo é uma opção de método de solução. A cada iteração do Método do Ponto Fixo é preciso resolver um sistema de equações lineares, e foi avaliado o uso do Método de Eliminação de Gauss com Pivotamento Parcial com Escala para realizar esta etapa dos cálculos. Os testes realizados mostraram que o Método de Eliminação de Gauss gera bons resultados. O uso de escala não afetou a qualidade das respostas. Também ficou evidente que é um método computacionalmente lento.

Keywords: Método de Eliminação de Gauss, Fluxo em Meio Poroso

1. Introdução

Ao longo de uma simulação de fluxo em meio poroso é preciso repetidamente resolver sistemas de equações não lineares. Diferentes métodos podem ser utilizados nesta resolução, como o método de Newton-Raphson ou o do Ponto Fixo. Ambos métodos precisam resolver sistemas de equações lineares a cada iteração.

Este trabalho avalia a aplicabilidade do Método de Eliminação de Gauss com Pivotamento Parcial com Escala para resolver o sistema de equações lineares de uma iteração do Método do Ponto Fixo. A avaliação é feita para um modelo de fluxo em meio poroso incompressível com duas fases (água e óleo).

2. Metodologia

2.1. Método de Eliminação de Gauss

Dado um sistema de n equações lineares na forma:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.1)$$

O Método de Eliminação de Gauss consiste em realizar sucessivas operações de soma e multiplicação com as

equações de forma a formar um novo conjunto de equações lineares equivalentes em forma triangular:

$$\begin{bmatrix} a'_{1,1} & a'_{1,2} & \dots & a'_{1,n} \\ 0 & a'_{2,2} & \dots & a'_{2,n} \\ \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & a'_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix} \quad (2.2)$$

Em geral, à exceção da primeira equação, os parâmetros a' e b' serão diferentes dos parâmetros a e b do sistema original, mas o vetor de respostas $\vec{x} = \{x_1, x_2, \dots, x_n\}$ será o mesmo. A vantagem da forma triangular é que os valores de x_i podem ser encontrados por substituição regressiva (de x_n para x_1):

$$x_i = \frac{b'_i - \sum_{j=i+1}^n a'_{i,j} x_j}{a'_{i,i}}, \text{ com } i = n, n-2, \dots, 1 \quad (2.3)$$

O processo de eliminação de Gauss, para transformar o sistema de equações original em um sistema de forma triangular, é feito de maneira sucessiva, anulando os valores da i -ésima coluna de todas as linhas abaixo da i -ésima linha. O algoritmo de eliminação de Gauss segue a seguinte lógica:

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} a_{k,j}^{(k)} \quad (2.4a)$$

$$b_i^{(k+1)} = b_i^{(k)} - \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}} b_k^{(k)} \quad (2.4b)$$

*Relatório número 10 como parte dos requisitos da disciplina IM253: Métodos Numéricos para Fenômenos de Transporte.

**Atualmente cursando doutorado no Departamento de Engenharia de Petróleo da Faculdade de Engenharia Mecânica da UNICAMP (Campinas/SP, Brasil).

Email address: t100675@dac.unicamp.br (Tiago C. A. Amorim)

com

$$\begin{aligned} k &= 1, 2, \dots, n-1 \\ i &= k+1, k+2, \dots, n \\ j &= k, k+1, \dots, n \end{aligned}$$

onde

$a_{i,j}^{(k)}$ é o valor da matriz de coeficientes na posição (i, j) após o k -ésimo passo¹

$b_i^{(k)}$ é o valor do vetor de constantes na posição (i) após o k -ésimo passo²

Como em 2.4 o termo $a_{k,k}^{(k)}$ (pivô) não pode ser nulo, pode ser necessário trocar a ordem de algumas das equações. Esta operação é o pivotamento parcial. Mesmo quando o termo não é nulo, é uma boa estratégia fazer o pivotamento para reduzir erros de arredondamento (e.g.: $a_{k,k}^{(k)}$ muito pequeno). Diferentes estratégias podem ser utilizadas para escolher que equações que devem trocar de lugar (em ordem de menos para mais sofisticada):

1. Eliminação de Gauss com Substituição Regressiva

Quando $a_{k,k}^{(k)} = 0$, trocar com a *próxima* linha (menor p , tal que $k \leq p \leq n$) com $a_{p,k}^{(k)} \neq 0$.

2. Eliminação de Gauss com Pivotamento Parcial

Trocar com a *próxima* linha que maximiza o módulo do pivô: $|a_{p,k}^{(k)}| = \max_{k \leq i \leq n} |a_{i,k}^{(k)}|$

3. Eliminação de Gauss com Pivotamento Parcial com Escala

Trocar com a *próxima* linha que maximiza o módulo do pivô com relação aos elementos da sua linha:

$$\frac{|a_{p,k}^{(k)}|}{\max_{1 \leq j \leq n} |a_{p,j}|} = \max_{k \leq i \leq n} \frac{|a_{i,k}^{(k)}|}{\max_{1 \leq j \leq n} |a_{i,j}|}$$

Se, após a troca, $a_{k,k}^{(k)}$ continuar nulo, o sistema não tem resposta única. O algoritmo deve retornar que não consegue resolver o sistema de equações.

Resumindo, o Método de Eliminação de Gauss se resume a:

1. Da primeira à penúltima linha ($k = 1, 2, \dots, n-1$):
 - (a) Realizar pivotamento em função de algum dos critérios enumerados acima (1,2 ou 3).
 - (b) Se $a_{k,k}^{(k)} = 0$, emitir mensagem de erro e sair do código.
 - (c) Aplicar 2.4 aos elementos das linhas seguintes ($i = k+1, k+2, \dots, n$).
2. Aplicar 2.3 e retornar o vetor de respostas $\vec{x} = \{x_1, x_2, \dots, x_n\}$.

Como o método realiza, a cada passo, operações em todas as linhas abaixo da linha atual, é de se esperar que as operações necessárias cresçam muito rapidamente com o número de variáveis³.

¹Seguindo esta notação: $a'_{i,j} = a_{i,j}^{(i-1)}$.

²De forma correspondente: $b'_i = b_i^{(i-1)}$.

³Maiores discussões em [1].

2.2. Fluxo Bifásico Incompressível em Meio Poroso

As equações que regem o problema de fluxo em meio poroso derivam fundamentalmente da equação de conservação de massa 2.5 e da equação de fluxo em meio poroso 2.6, a lei de Darcy [2].

$$\sum_p \nabla \cdot (y_{cp} \rho_p v_p) + \sum_p (y_{cp} \rho_p q_p) + \sum_p \frac{\partial}{\partial t} (\phi y_{cp} \rho_p S_p) = 0 \quad (2.5)$$

$$v_p = -\frac{kk_{rp}}{\mu_p} \left(\frac{\partial p_p}{\partial x} - \gamma_p \frac{\partial D}{\partial x} \right) = -\frac{kk_{rp}}{\mu_p} \left(\frac{\partial \Phi_p}{\partial x} \right) \quad (2.6)$$

A maioria dos simuladores de fluxo comerciais utilizam diferenças finitas [3][4]. Podemos acoplar as equações 2.5 e 2.6, e aplicar uma discretização no tempo (Δt) e no espaço (Δx) para um volume de controle ($V = \Delta x \Delta y \Delta z$)⁴. A forma unidimensional desta equação discretizada é apresentada a seguir 2.7.

$$\begin{aligned} & \sum_p \left(\frac{\Delta y \Delta z}{\Delta x} y_{cp} \rho_p \frac{kk_{rp}}{\mu_p} \right)_{i+\frac{1}{2}} (\Phi_{p,i+1} - \Phi_{p,i}) \\ & + \left(\frac{\Delta y \Delta z}{\Delta x} y_{cp} \rho_p \frac{kk_{rp}}{\mu_p} \right)_{i-\frac{1}{2}} (\Phi_{p,i-1} - \Phi_{p,i}) \\ & + q_{cp}^w = \frac{1}{\Delta t} \sum_p (V \phi y_{cp} \rho_p S_p)_i^{t_i+\Delta t} - (V \phi y_{cp} \rho_p S_p)_i^{t_i} \end{aligned} \quad (2.7)$$

Na sua forma mais geral o problema de fluxo em meio poroso precisa ser resolvido para cada um dos componentes que constituem as fases envolvidas. É comum utilizar uma abordagem simplificada, em que poucos componentes são utilizados para representar os fluidos envolvidos. Esta simplificação é usualmente aplicada quando as trocas de fase são *bem comportadas* e passíveis de serem representadas por um conjunto de tabelas, em substituição às equações de estado. Esta abordagem é conhecida como *Black-Oil*.

As equações também se simplificam ao substituir as densidades (ρ_p) e concentrações de componentes (y_{cp}) pelo fator volume de formação da fase (B) e as relações volumétricas entre as fases e os componentes (R)⁵.

Quando apenas água e óleo estão envolvidos na simulação, apenas duas equações de conservação de massa são necessárias. Como a soma das saturações é igual à unidade ($S_w + S_o = 1$) e negligenciando a tensão interfacial entre a água e o óleo ($p_o = p_w = p$), para cada célula é preciso resolver apenas duas variáveis: p e S_w :

⁴Usualmente chamado de célula.

⁵Ver dedução completa em [2].

Tabela 1: Formato da matrix de coeficientes para modelo unidimensional.

$$\begin{aligned} & \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_w \right)_{i+\frac{1}{2}} (p_{i+1} - p_i - \gamma_w \Delta D_{i+\frac{1}{2}}) \\ & + \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_w \right)_{i-\frac{1}{2}} (p_{i-1} - p_i - \gamma_w \Delta D_{i-\frac{1}{2}}) \\ & = \frac{1}{\Delta t} \left(\frac{V \phi S_w}{B_w} \right)_i^{t_i+\Delta t} - \left(\frac{V \phi S_w}{B_w} \right)_i^{t_i} + q_w^{std} \end{aligned} \quad (2.8)$$

$$\begin{aligned} & \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_o \right)_{i+\frac{1}{2}} (p_{i+1} - p_i - \gamma_o \Delta D_{i+\frac{1}{2}}) \\ & + \left(\frac{\Delta y \Delta z}{\Delta x} \lambda_o \right)_{i-\frac{1}{2}} (p_{i-1} - p_i - \gamma_o \Delta D_{i-\frac{1}{2}}) \\ & = \frac{1}{\Delta t} \left(\frac{V \phi (1 - S_w)}{B_o} \right)_i^{t_i+\Delta t} - \left(\frac{V \phi (1 - S_w)}{B_o} \right)_i^{t_i} + q_o^{std} \end{aligned} \quad (2.9)$$

com:

$$\lambda_p = \frac{k k_{rp}}{B_p \mu_p}$$

Observa-se que a pressão (p) e a saturação de água (S_w) da i -ésima célula tem influência apenas na própria célula e nas células vizinhas. Desta forma, a matriz dos coeficientes deste problema tem termos não nulos em três bandas ao redor da diagonal principal (tabela 1⁶).

Na forma bi e tridimensional de 2.8 e 2.9 são adicionados termos de conexão com as células vizinhas nas direções j e k , utilizando termos equivalentes aos apresentados para a direção i . Na matriz de coeficientes aparecem novos valores não nulos fora da diagonal principal (tabela 2).

2.3. Modelo de Reservatório Proposto

O modelo de reservatório é plano, com forma quadrada e mesmo refinamento em ambas direções ($\Delta i = \Delta j$). Existe um injetor de água em um dos cantos (célula [1,1])

⁶Os quadrados em preto e cinza indicam os valores não nulos da matriz, associados, respectivamente, aos termos das equações com os parâmetros da própria célula e com os parâmetros das células vizinhas.

Tabela 2: Formato da matrix de coeficientes para modelo bidimensional.

e um produtor na diagonal oposta (célula $[n,n]$). As propriedades de rocha (ϕ e k) são constantes.

Em uma simulação *Black-Oil* com água e óleo, a simulação de fluxo parte de um conjunto de valores de p e S_w no tempo inicial, para cada uma das células, e uma série de controles de produção (q_p^{std}). O simulador precisa resolver o conjunto de equações não lineares 2.8 e 2.9 a cada passo de tempo.

A implementação mais conhecida para resolver este conjunto de equações não lineares é o Método de Newton-Raphson. Para sistemas incompressíveis (onde ϕ , B e μ são constantes) este conjunto de equações é *fracamente* não linear, e apenas os valores de k_{ro} e k_{rw} são função das variáveis de estado. O conjunto de equações resultantes pode ser resolvido adequadamente com o Método do Ponto Fixo. As equações são resolvidas como um conjunto de equações lineares de modo iterativo, isto é, utilizando o resultado de uma iteração como dado de entrada para resolver os termos k_{ro} e k_{rw} da próxima iteração, até que a convergência é atingida.

A proposta deste trabalho é avaliar a performance do Método de Eliminação de Gauss com Pivotamento Parcial com Escala. A partir de um código de fluxo em meio poroso bidimensional e incompressível⁷ foram gerados arquivos CSV com as matrizes de coeficientes e vetores de constantes para a resolução de uma iteração do Método do Ponto Fixo, para diferentes refinamentos de malha.

Também foram exportados os resultados de cada uma das respostas destes sistemas de equações. No simulador de fluxo a resolução do sistema de equações foi feito com o pacote Numpy. Segundo a documentação do pacote, são utilizadas rotinas do LAPACK [5].

3. Implementação

Todo o código utilizado nesta análise foi desenvolvido em C++. A duas funções principais são:

⁷O código deste simulador, desenvolvido em Python, encontra-se em <https://github.com/TiagoCAAmorim/IntegratedModel/>

readCSV Função que recebe um *string* com o caminho de um arquivo CSV e faz a sua leitura. É assumido que é utilizado vírgula como separador. A função retorna uma matrix de *double*.

SolveGauss Função que recebe uma matrix de *double* e um vetor de *double*, e resolve o sistema de equações lineares usando Eliminação de Gauss com Pivotamento Parcial. Existe a opção de realizar o pivotamento com e sem uso de escala.

4. Resultados

Foram realizadas duas avaliações. A primeira foi da qualidade das respostas encontradas com o Método de Eliminação de Gauss. As respostas da implementação do Método de Eliminação de Gauss foram comparadas com as respostas do método de solução de sistema de equações lineares utilizado no simulador de fluxo em meio poroso (do pacote Numpy). Foram feitos testes com as duas variações do Método de Eliminação de Gauss: sem e com uso de escala.

A medida da qualidade das respostas de cada método foi baseada no residual na forma $R = Ax - B$, onde A é a matrix de coeficientes, B é o vetor de constantes e x é o vetor da solução do sistema linear $Ax = B$. Foram avaliadas duas normas do vetor residual: L^2 (4.1) e L^∞ (4.2).

$$\|x\|_2 := \sqrt{\sum_i x_i^2} \quad (4.1)$$

$$\|x\|_\infty := \max_i |x_i| \quad (4.2)$$

As figuras 1 e 2 comparam as normas dos residuais da solução dos sistemas de equações propostos com três métodos:

- Eliminação de Gauss com Pivotamento Parcial
- Eliminação de Gauss com Pivotamento Parcial com Escala
- Rotinas do Numpy⁸

Observa-se que para o problema proposto de resolver sistemas de equações de um modelo de fluxo em meio poroso bidimensional, não há necessidade de usar escala no Método de Eliminação de Gauss. Uma segunda constatação é de que a qualidade das respostas geradas com o Método de Eliminação de Gauss é tão boa quanto a dos métodos numéricos implementados no Numpy, que seguramente são mais sofisticados que os métodos implementados para esta avaliação.

⁸Este residual foi calculado com base nas respostas geradas com as rotinas de cálculo utilizadas no simulador de fluxo. Estes resultados fazem parte desta comparação para representar respostas de *melhor qualidade*.

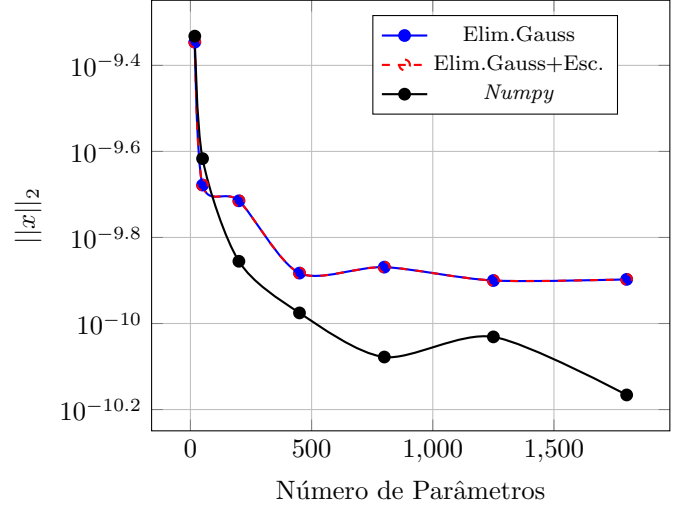


Figura 1: Norma L^2 do residual das soluções de sistemas de equações lineares com diferentes métodos.

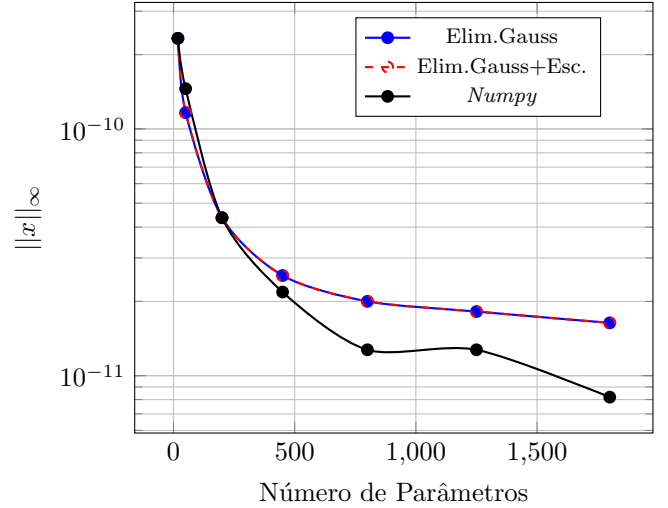


Figura 2: Norma L^∞ do residual das soluções de sistemas de equações lineares com diferentes métodos.

O Método de Eliminação de Gauss com Pivotamento Parcial com Escala gera resultados de boa qualidade, mesmo para sistemas com um grande número de variáveis (figura 3), mas, como esperado, é um método lento. A figura 4 mostra que o tempo necessário para resolver cada um dos sistemas de equações propostos cresce rapidamente. Um modelo de fluxo bidimensional com 80 divisões em cada direção tem um total de $80 \times 80 \times 2 = 12.800$ variáveis. A solução de uma iteração deste modelo tomou mais de 6 horas para ser completada⁹.

O código foi implementado em C++ e em um único arquivo. Pode ser encontrado em <https://github.com/TiagoCAAmorim/numerical-methods>.

⁹Testes realizados com uma CPU Intel Core i7-8550U.

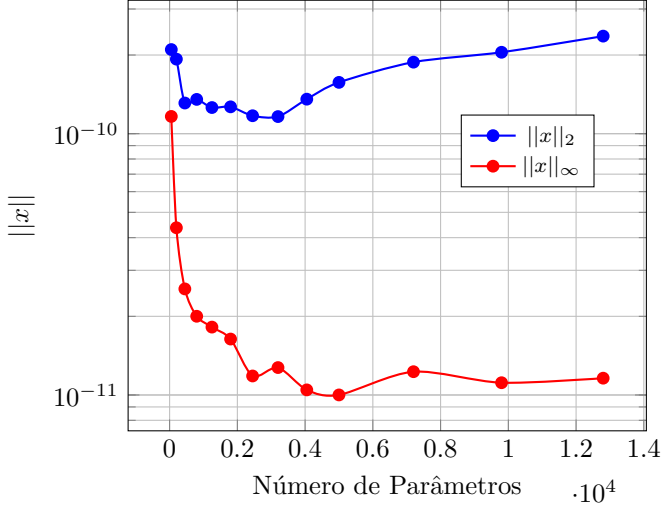


Figura 3: Normas do residual das soluções dos sistemas de equações lineares resolvidos com o Método de Eliminação de Gauss com Pivotamento Parcial com Escala.

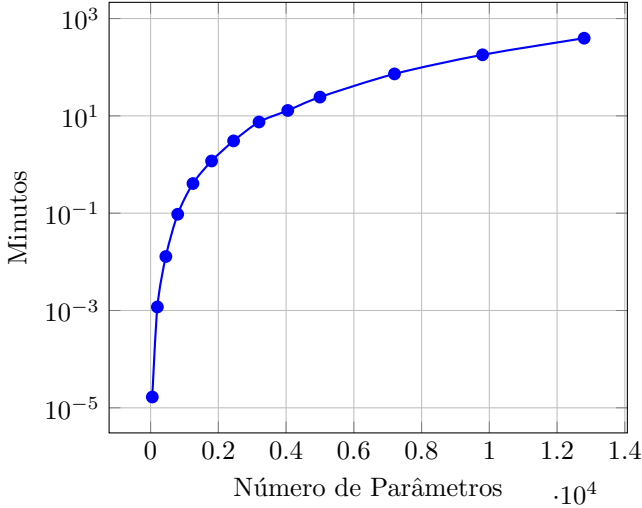


Figura 4: Tempo necessário para resolver sistemas de equações com o Método de Eliminação de Gauss com Pivotamento Parcial com Escala.

5. Conclusão

O Método de Eliminação de Gauss com Pivotamento Parcial com Escala gerou boas respostas para os sistemas de equações lineares testados. O residual das respostas do método implementado foram da mesma ordem de grandeza dos resultados alcançados com um pacote de computação científica popular. O Método de Eliminação de Gauss se mostrou computacionalmente lento, chegando a ser inviável para resolver sistemas de equações com um número maior de variáveis.

Apêndice A. Lista de Variáveis

B_o : Fator volume de formação do óleo no reservatório (m^3/m^3).

B_w : Fator volume de formação da água no reservatório (m^3/m^3).

D : Profundidade.

Δx : Discretização espacial na direção i .

Δy : Discretização espacial na direção j .

Δz : Discretização espacial na direção k .

Δt : Discretização temporal.

γ_p : Peso específico da fase p ($\gamma_p = \rho_p g$).

k : Permeabilidade absoluta do meio poroso.

k_{rp} : Permeabilidade relativa da fase p .

μ_p : Viscosidade da fase p .

p_p : Pressão da fase p .

ϕ : Porosidade da rocha.

q_p : Vazão volumétrica da fase p .

q_{cp}^w : Vazão mássica do componente c na fase p .

q_{cp}^{std} : Vazão volumétrica da fase p medida em condições padrão (*standard*).

ρ_p : Densidade da fase p .

S_p : Saturação da fase p no meio poroso.

V : Volume total do volume de controle (célula).

v_p : Velocidade da fase p .

y_{cp} : Concentração do componente c na fase p .

Referências

- [1] R. L. Burden, J. D. Faires, A. M. Burden, *Análise numérica*, Cengage Learning, 2016.
- [2] L. P. Dake, *Fundamentals of reservoir engineering*, Elsevier, 1983.
- [3] C. M. G. L. (CMG), *Cmg imex user's manual*, CMG Calgary, Canada, 2022.
- [4] E. U. M. SCHLUMBERGER, *Technical description*, Schlumberger Ltd (2009) 519–538.
- [5] J. J. Dongarra, J. W. Demmel, S. Ostrouchov, *Lapack: a linear algebra library for high-performance computers*, in: *Computational Statistics: Volume 1: Proceedings of the 10th Symposium on Computational Statistics*, Springer, 1992, pp. 23–28.