

Eficiência dos Métodos Multigrid Algébricos para a Solução da Equação do Fluxo Livre Estacionário em Domínio Georreferenciado

J. P. M. SANTOS^{1*}, A. FIRMIANO², H. C. JHUNIOR³ e E. WENDLAND³⁴

Recebido em 23 de dezembro de 2021 / Aceito em 21 de abril de 2022

RESUMO. Neste artigo o método multigrid algébrico baseado em agregação suavizada, o método clássico de Ruge-Stuben e o método *GMRES* pré-condicionado por multigrid algébrico foram utilizados para a solução da equação do fluxo livre estacionário em domínio georreferenciado. A disponibilidade dos códigos computacionais permitiu avaliar a aproximação de elementos finitos sob a perspectiva dos métodos multigrid algébricos e respectiva combinação, como pré-condicionante, com o método *GMRES*. As diferenças máximas entre soluções por diferentes métodos, o tempo necessário para obter as soluções dos sistemas lineares associados, em cada uma das iterações de Picard, os residuais de cada um dos métodos iterativos e os residuais em cada uma das iteradas de Picard são apresentados e discutidos. Como resultado da análise, os métodos pré-condicionados são mais eficientes no sentido do menor tempo computacional aliado à estabilidade do número de iterações. A análise dos residuais das iterações de Picard permite comparar a evolução dos diferentes métodos de solução dos sistemas lineares. O detalhamento dos residuais dos métodos iterativos em cada passo das iterações de Picard permitiu uma visão mais abrangente e uma análise da convergência. Em detalhes, o método baseado em agregação suavizada necessita de um número expressivamente menor de iterações quando comparado ao método clássico de Ruge Stüben nas primeiras iterações de Picard. O pré-condicionamento reduz o número de iterações em relação às iterações iniciais e há uma persistência da redução do número de iterações do método baseado em agregação em relação ao método clássico.

Palavras-chave: elementos finitos, FEniCs, Python, métodos iterativos.

1 INTRODUÇÃO

Os modelos computacionais para previsão do comportamento de contaminantes em meio poroso saturado dependem das equações governantes de processos físicos, tais como dispersão,

*Corresponding author: João Paulo Martins dos Santos – E-mail: jp2@alumni.usp.br

¹Academia da Força Aérea, Pirassununga, SP, Brasil – E-mail: jp2@alumni.usp.br <https://orcid.org/0000-0002-0957-7119>

²Academia da Força Aérea, Pirassununga, SP, Brasil – E-mail: lezandro@gmail.com <https://orcid.org/0000-0002-7073-2261>

³Escola de Engenharia de São Carlos, EESC-USP, São Carlos, SP, Brasil – E-mail: correiajhr@gmail.com <https://orcid.org/0000-0003-3512-3943>

⁴Escola de Engenharia de São Carlos, EESC-USP, São Carlos, SP, Brasil – E-mail: ew@sc.usp.br <https://orcid.org/0000-0003-3374-608X>

difusão e advecção, químicos e biológicos envolvidos no fluxo do fluido. Estes podem ser traduzidos matematicamente em equações diferenciais, solucionadas por métodos numéricos ou analíticos. Um dos principais elementos da equação do transporte e nos processos de difusão no meio poroso é o campo de velocidades, o qual determina as trajetórias preferenciais para o contaminante. O campo de velocidade de um fluido é uma variável que pode ser definida em função das coordenadas espaciais e do tempo, influenciando na dispersão e na classificação do regime de escoamento.

Assim, a adequada determinação do campo de velocidades é essencial ao processo de simulação. Ao integrar um Sistema de Informação Geográfica (SIG), o domínio irregular delimitado por um conjunto discreto de pontos, as condições de fronteira e os demais elementos presentes na equação do fluxo ocasionam novas dificuldades técnicas, tanto para a geração da malha de elementos finitos, quanto para a definição adequada das condições de fronteira. Além disso, é necessário, a adequada caracterização dos parâmetros de condutividade, poços de bombeamento e zonas de recarga. Finalmente, há a necessidade de encontrar uma solução para os sistemas algébricos que surgem da utilização dos métodos de aproximação, tais como diferenças finitas ou elementos finitos.

Neste contexto, o artigo foca na utilização dos métodos multigrid algébricos para a solução dos sistemas lineares associados às iterações do Método de Picard, o qual foi originado da aproximação de elementos finitos da equação do fluxo livre estacionário. Especificamente, os métodos clássico de Ruge-Stüben (RS), o método baseado em agregação suavizada (SA) e o método *GMRES* pré-condicionado por ambos *RS*, *GMRES_{RS}*, e *SA*, *GMRES_{SA}*, foram comparados. O tempo de solução dos sistemas lineares associados às iterações de Picard, os residuais dos métodos iterativos em cada uma das iterações de Picard e o perfil dos resíduos das iterações de Picard foram obtidos para todos os métodos citados e os resultados são analisados em detalhe.

2 EQUAÇÃO MODELO

Neste artigo, o modelo matemático que determina a distribuição de cargas hidráulicas h em meio poroso saturado sob a influência de regiões de recarga W e de poços de bombeamento f é dada por (Ver [2] para maiores detalhes):

$$-\nabla(Kh \cdot \nabla h) + W = f, \quad \text{em } \Omega \quad (2.1)$$

em que K representa a condutividade hidráulica do meio e Ω é o domínio computacional. A imposição das condições de fronteira do tipo Neumann, Dirichlet ou ambas, possibilita obter a solução $h = h(x, y)$, que fornece a distribuição de cargas hidráulicas sobre o domínio e, por meio da *Lei de Darcy*, o campo de velocidades $v = (v_x, v_y)$.

A aproximação de elementos finitos da equação (2.1), obtida por meio da formulação variacional, gera um sistema algébrico de equações em que a função $h = h(x, y)$ agora representa a solução de elementos finitos. Ao utilizar o método de aproximações sucessivas de Picard, o sistema não-

linear, da forma $A(h) \cdot h = b$, é convertido em uma sequência de sistemas lineares, denotados convenientemente por:

$$A(h^k) \cdot h^{k+1} = b^{k+1}, k = 1, 2, \dots, N_k, \quad (2.2)$$

em que as sucessivas aproximações h^1, h^2, \dots, h^{N_k} foram obtidas por meio de métodos diretos ou iterativos e h^0 é uma aproximação inicial para o método de Picard. Um detalhamento da formulação matemática da formulação variacional pode ser encontrada em [6], [10], [17].

3 MÉTODO MULTIGRID ALGÉBRICO

A solução de elementos finitos de 2.1 é $h = h^{N_k}$, solução de 2.2, e tal que a norma das diferenças $\|h^{N_k} - h^{N_k-1}\| < \varepsilon$.

Ao utilizar um método direto para obter as soluções $h^k, k = 1, 2, \dots, N_k$, os erros de solução são aqueles provenientes da aritmética de ponto flutuante. Por outro lado, um método iterativo calcula uma solução aproximada h^{k*} por meio de uma sequência de soluções $h_0^{k*}, h_1^{k*}, \dots, h_{N_k}^{k*}$. O critério de parada, em geral, está atrelado a norma da diferença de duas aproximações sucessivas $\|h_{N_k}^{k*} - h_{N_k-1}^{k*}\|$. Uma discussão detalhada sobre os métodos iterativos pode ser encontrada em [1].

Dentre as possibilidades de escolha dos métodos iterativos estão os métodos multigrid. Estes podem ser classificados em geométricos ou algébricos e são diferenciados pela respectiva forma estrutural de solução do sistema linear. Segundo Stüben, [3], no método geométrico a estratégia que gera a sequência de malhas é fixa e as interpolações dos resultados entre as malhas têm por base a estrutura geométrica da sequência. Por outro lado, o método algébrico assume um método iterativo (Gauss-Seidel) e têm objetivo na construção dos operadores de transferência e restrição com foco nos coeficientes da matriz do sistema linear. A construção dos operadores de restrição e prolongamento, a qual é dependente das hipóteses adotadas sobre a representação do erro na malha grossa, conduz aos métodos clássicos de Ruge-Stüben e aos métodos baseados em agregação (Ver [3] para mais detalhes). Estes, ao adicionar um processo de suavização dos operadores, fornecem os métodos multigrid baseados em agregação suavizada (SA). Discussões mais detalhadas sobre os métodos multigrid podem ser encontradas em [3], [5], [16], [4].

4 IMPLEMENTAÇÃO

O domínio georreferenciado, a geração da malha de elementos finitos, a condição de fronteira de Dirichlet e as coordenadas dos poços juntamente com os respectivos valores discretos de carga hidráulica são provenientes de [8]. As informações podem ser complementadas por meio das referências [7] e [9].

Considerando a metodologia do Projeto FEniCS [13], a qual é baseada na utilização das formas linear e bilinear originadas da formulação variacional, a implementação do método de elementos finitos pode ser executada. O código computacional Python^R para a equação modelo (2.1) e iteradas de Picard com Fatoração LU é disponibilizado em [14] e apresentado na Figura 1.

```

1 u = TrialFunction(V); v = TestFunction(V)
2 u_k = interpolate(WellsLocFunc, V) ; f = Expression('W',W=W)
3 a = inner(K*q(u_k)*nabla_grad(u), nabla_grad(v))*dx
4 L = f*v*dx-g*v*ds
5 A = assemble(a); b = None
6 u = Function(V)
7 eps = 1.0; tol = 1.0E-5; iter = 0; maxiter = 200
8 while eps > tol and iter < maxiter:
9     iter += 1
10    solve(a == L, u, bc)
11    diff = u.vector().array() - u_k.vector().array()
12    eps = np.linalg.norm(diff, ord=np.Inf)
13    u_k.assign(u)

```

Figura 1: Código Python para implementação do método de elementos finitos seguindo metodologia FEniCS [13]. Fonte: [14].

Os critérios de parada estabelecidos para as iterações de Picard foram o valor $\varepsilon = \max |u^{k+1} - u^k|$ ou um máximo de iterações [13]. Neste artigo, o número máximo de iterações de Picard permitido foi $maxiter = 400$ e tolerância $\varepsilon = 10^{-6}$. A estimativa inicial do método de Picard, h^0 , foi aquela obtida por meio de funções de bases radiais com dados da fronteira e poços de observação dentro do domínio computacional, conforme delineado em [8]. Por sua vez, a estimativa inicial, h_0^{k*} , para os métodos iterativos, na iteração k do método de Picard, foram aquelas provenientes da aproximação de Picard da iteração $k - 1$. Esta estratégia é baseada na hipótese de que a distribuição de cargas h^{k-1} é uma estimativa mais adequada do que uma estimativa aleatória ou nula.

A disponibilidade dos dados e dos códigos permitiu inserir modificações para considerar um método iterativo para a solução dos sistemas algébricos gerados. O método *GMRES* é proveniente do pacote *linalg*, disponível em SciPy [18], enquanto que os métodos multigrid *RS* e *SA* são provenientes de PyAMG [12]. Para fins de comparação, todos os métodos iterativos utilizaram os mesmos critérios de parada: tolerância $\varepsilon = 10^{-12}$ e número de iterações $maxiter = 10000$.

Os resultados gráficos utilizaram os recursos disponíveis em FEniCS [13] e na biblioteca Matplotlib [11], enquanto que as manipulações dos vetores de soluções numéricas foram realizadas por meio da biblioteca NumPy [15].

5 RESULTADOS

Inicialmente, o sistema de equações algébricas, obtido da aproximação do equacionamento do fluxo livre e estacionário, foi resolvido com o método *SA* e com o método *RS*. Em seguida os métodos *GMRES_{RS}* e *GMRES_{SA}* foram utilizados. O código computacional foi modificado de modo a inserir um termo de recarga na região retangular $|x - 12000.0| < 1500.0$ e $|y - 15000.0| \leq 1000.0$.

A superfície que fornece a distribuição de cargas hidráulicas $h = h(x, y)$ é apresentada na Figura 2 por meio das ferramentas disponíveis em FeniCS [13]. A aplicação da *Lei de Darcy* à função h gera o campo de velocidades de água subterrânea. A sobreposição das linhas equipotenciais ou curvas de nível da função h ao campo de vetores normalizados, representando o campo de velocidades, são apresentados na Figura 3. Neste caso, o campo de vetores é apresentado por meio de uma amostra de pontos da malha e respectivos valores numéricos da *Lei de Darcy* de modo a fornecer uma visão geral do campo de velocidades e ainda estabelecer um equilíbrio na quantidade de vetores nas regiões de baixa densidade de pontos da malha. Ressalta-se que a utilização de todos os pontos ocasiona dificuldades para a visualização da estrutura vetorial e das linhas equipotenciais. Ambas as Figuras 2 e 3 contém informações que favorecem a análise qualitativa da função h e, por consequência, o comportamento do escoamento. A função h possibilita identificar as áreas de recarga e áreas de rebaixamento e o delineamento da estrutura do campo de velocidades.

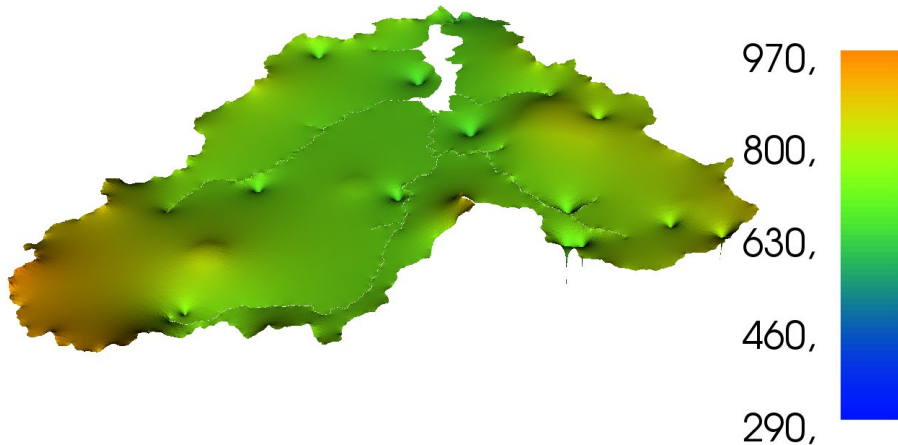


Figura 2: Solução numérica da equação do fluxo em domínio georreferenciado. Fonte: Modificado de [8] para englobar um termo de recarga adicional.

Por sua vez, a Figura 3 identifica a direção do escoamento e permite avaliar mais detalhes do respectivo comportamento. De posse dos resultados gráficos é possível inferir o deslocamento de uma substância presente no meio poroso, analisar a influência do conjunto de poços ou zonas de recarga sobre o fluxo subterrâneo. Ressalta-se que estes resultados diferem daqueles apresentados em [8] devido à presença do termo de recarga.

A seguir são apresentados alguns resultados numéricos do processo de solução dos sistemas lineares associados à obtenção da solução de elementos finitos h . É necessário observar que parâmetros idênticos para o número máximo de iterações e tolerância foram adotados tanto para o método de Picard quanto para os métodos iterativos; os parâmetros multigrid dos métodos RS e SA são mantidos idênticos para os respectivos métodos $GMRES_{RS}$ e $GMRES_{SA}$.

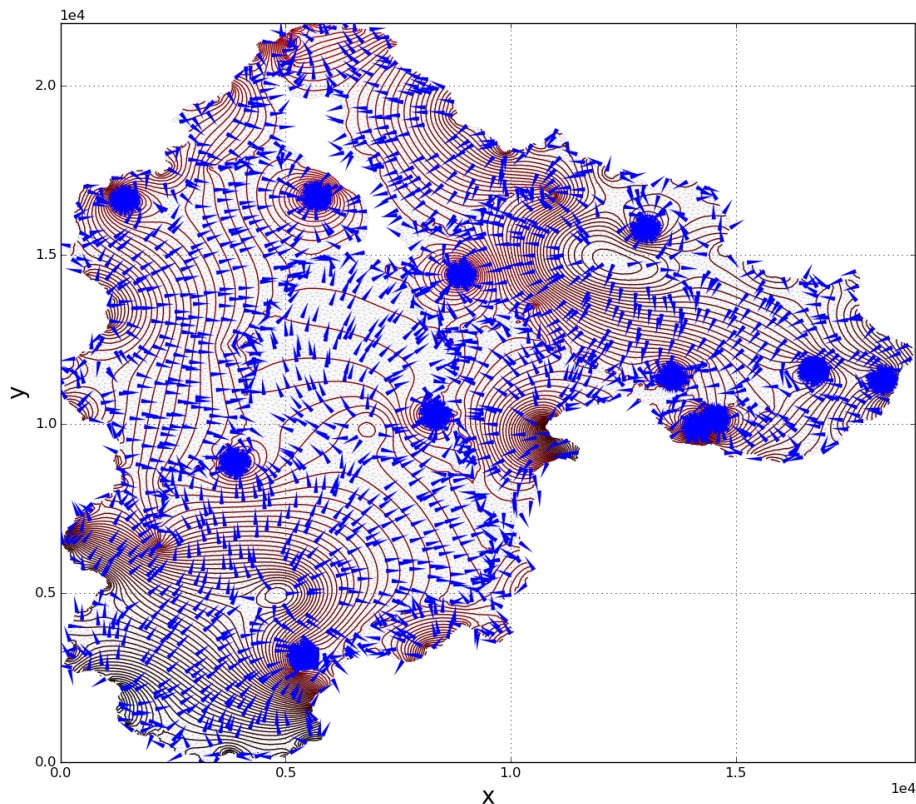


Figura 3: Campo de velocidades e curvas de nível da solução numérica da equação do fluxo em domínio georreferenciado. Fonte: Modificado de [8] para englobar um termo de recarga adicional.

As Figuras 4 e 5 mostram os respectivos parâmetros multigrid, sendo que a descrição de cada elemento pode ser encontrada em [12].

```

1 mls = smoothed_aggregation_solver(A,B=B0, symmetry='symmetric', strength='
  symmetric',
2   aggregate='standard', smooth=('jacobi', {'omega': 4.0/3.0}),
3   presmooher=('block_gauss_seidel',{'sweep': 'symmetric'}),
4   postsmooher=('block_gauss_seidel',{'sweep': 'symmetric'}),
5   improve_candidates=[('block_gauss_seidel',{'sweep': 'symmetric',
  iterations': 4}), None],
6   max_levels = 10, max_coarse = 1000, coarse_solver='lu',
  diagonal_dominance=False)

```

Figura 4: Parâmetros para o método multigrid algébrico baseado em agregação suavizada (SA).

```

1  mls = ruge_stuben_solver(Asp,
2      strength=('classical', {'theta': .25}),
3      CF='RS',
4      presmoothen=('block_gauss_seidel', {'sweep': '
symmetric'}),
5      postsmoothen=('block_gauss_seidel', {'sweep': '
symmetric'}),
6      max_levels=10,
7      coarse_solver='lu',
8      max_coarse=1000)

```

Figura 5: Parâmetros para o método multigrid algébrico clássico (RS).

A análise dos resultados inicia-se na comparação das soluções numéricas obtidas. Considere que h_{Lu} , h_{RS} , h_{SA} , $h_{GmresRS}$ e $h_{GmresSA}$ são as soluções de elementos finitos obtidas por meio dos métodos da fatoração LU e métodos RS , SA , $GMRES_{RS}$ e $GMRES_{SA}$, respectivamente. A análise qualitativa das soluções, que correspondem àquela obtida após atingir o critério de parada para as iterações de Picard, não mostrou evidência significativa para as diferenças, as quais podem ser confirmadas pelas máximas diferenças entre as soluções: $\max |h_{Lu} - h_{RS}| = 1.67 \cdot 10^{-4}$, $\max |h_{Lu} - h_{SA}| = 3.32 \cdot 10^{-06}$, $\max |h_{Lu} - h_{GmresRS}| = 2.96 \cdot 10^{-11}$, $\max |h_{Lu} - h_{GmresSA}| = 3.32 \cdot 10^{-06}$, $\max |h_{RS} - h_{GmresRS}| = 1.67 \cdot 10^{-4}$, $\max |h_{SA} - h_{GmresSA}| = 0.00$ e $\max |h_{GmresRS} - h_{GmresSA}| = 3.32 \cdot 10^{-6}$.

O comportamento ou evolução dos resíduos de todas as soluções das iterações de Picard são ilustrados na Figura 6. Os valores são visualmente idênticos nas primeiras iterações, mas há uma aparente divergência a partir da décima iteração. A aparente divergência no número de iterações é consequência do valor nulo obtido na última iteração de Picard para os métodos SA e RS , o qual não pode ser mostrado em escala logarítmica. Em outras palavras, os métodos RS e SA não são capazes de introduzir nova informação entre duas soluções sucessivas do método de Picard, ou seja, o processo iterativo atinge um estado estacionário.

A Figura 7 mostra os tempos necessários para calcular as soluções dos sistemas lineares, gerados pela aproximação de elementos finitos, em cada iteração do método de Picard. Como consequência, segue que o método RS requer cerca de 7 vezes mais tempo na primeira iteração com redução dessa magnitude nos passos subsequentes. Apesar do comportamento decrescente, o menor tempo é equivalente, ao menos aproximadamente, ao tempo dos métodos SA e $GMRES_{SA}$. Por sua vez, os métodos SA e $GMRES_{SA}$ possuem tempos aproximadamente iguais durante todo o processo. Finalmente, o método $GMRES_{RS}$ segue o mesmo padrão que o método $GMRES_{SA}$, mas com um tempo ligeiramente maior. O maior tempo gasto pelo método RS em relação ao respectivo pré-condicionado está ligado ao número de iterações necessárias para a convergência, conforme apresentado na sequência e em conjunto com a análise dos residuais dos métodos iterativos para todas as iterações de Picard.

As Figuras 8 e 9 ilustram a redução residual relativa dos métodos multigrid para a primeira iteração do método de Picard. Neste caso, os residuais são aqueles provenientes da aplicação

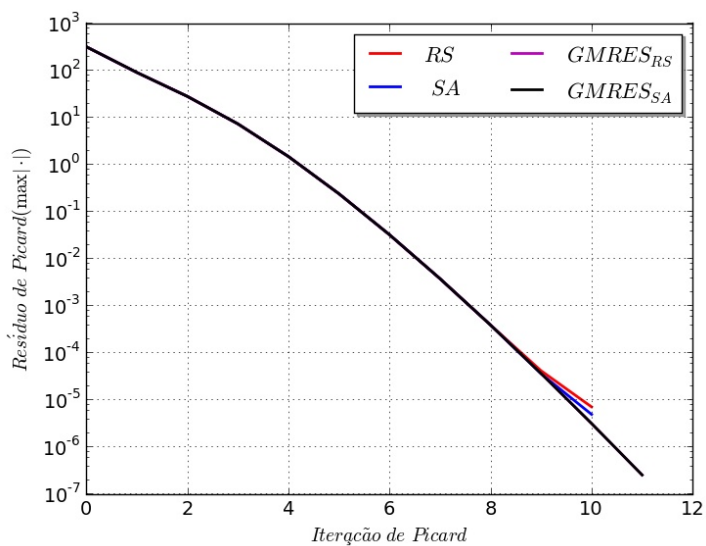


Figura 6: Resíduos em cada uma das iterações do método de Picard em escala logarítmica no eixo Y.

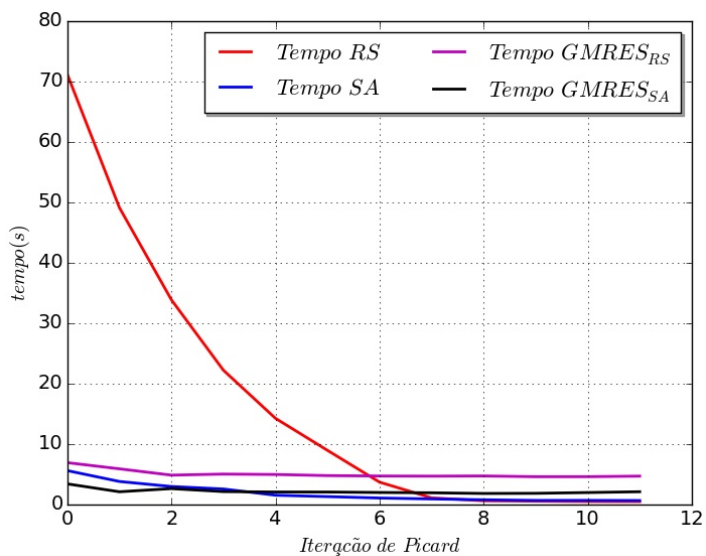


Figura 7: Tempo requerido em cada iteração do método de Picard para o método multigrid algébrico baseado em agregação, SA, o método multigrid clássico de Ruge-Stüben, RS, e o método GMRES pré-condicionado por SA e RS.

sucessiva do processo iterativo dos métodos multigrid algébricos SA e RS e, respectivamente, dos pré-condicionados $GMRES_{RS}$ e $GMRES_{SA}$.

Da análise da Figura 8 resulta que o número de iterações necessárias para a convergência do método SA é muito menor do que aquele requerido pelo método RS . Uma consequência do maior número de iterações é o tempo associado à solução do sistema linear. Neste caso, tanto os parâmetros do método RS quanto do método SA foram escolhidos arbitrariamente e sem qualquer tipo de procedimento para otimização. Deve ser ressaltado que ambos os métodos são influenciados pela escolha dos parâmetros e, portanto, configurações distintas podem afetar a performance de ambos tanto no número de iterações quanto no processo de convergência em si devido ao surgimento de valores não numéricos. Os respectivos parâmetros adotados são aqueles apresentados nas Figuras 4 e 5.

Uma argumentação similar pode ser feita ao comparar os resultados mostrados na Figura 9. Ainda é evidente que o número de iterações necessárias para a convergência do método $GMRES_{SA}$ é muito menor que aquele requerido pelo método $GMRES_{RS}$, porém a magnitude da diferença entre os números de iterações é menor. Também há um menor tempo associado à solução do sistema linear, mas a diferença temporal não é tão expressiva como no caso anterior dos métodos RS e SA . A comparação entre SA , $GMRES_{SA}$ e RS , $GMRES_{RS}$ também direciona para a conclusão de que o método pré-condicionado executa menos iterações. Por fim, a comparação entre os métodos pré-condicionados favorece o método SA ao levar em conta o número de iterações, o tempo e o residual.

Por fim, os resultados dos residuais das iterações dos métodos multigrid, nas iterações de Picard $iter = 0, 1, \dots, 9$, são apresentadas nas Figuras 10, 11, 12 e 13. Dessa forma, os resultados analisados na Figura 8 correspondem à iteração $iter = 0$ nas Figuras 10 e 11, enquanto que os resultados analisados na Figura 9 correspondem à iteração $iter = 0$ nas Figuras 12 e 13.

A análise conjunta das quatro figuras não permite afirmar um comportamento qualitativo similar àquele apresentado nas Figuras 8 e 9 para todas as iteradas de Picard, pois o número de iterações dos métodos RS e SA decrescem a uma unidade na última iteração, que é quando não há mudança de valores entre as soluções na iteração de Picard de número 10 e 11. Dessa forma, os resultados da Figura 7 podem ser justificados de forma consistente, ou seja, a redução do tempo do método RS está associada à redução no número de iterações executadas. Por sua vez, a análise do método SA em conjunto com o número de iterações executadas mostra o custo computacional de uma iteração é maior que o custo de uma iteração do método RS . Em palavras, até a sétima iteração de Picard, o método RS executa uma quantidade extra de iterações que compensa o custo computacional. A partir deste ponto, o tempo gasto por RS em $iter_6 = 17$ é menor que o tempo gasto por SA para executar $iter_6 = 9$. Em ambos os casos, a redução do número de iterações parece estar diretamente associada ao fato de ocorrer uma diferença nula nas duas últimas iterações de Picard e tal assunto não foi explorado.

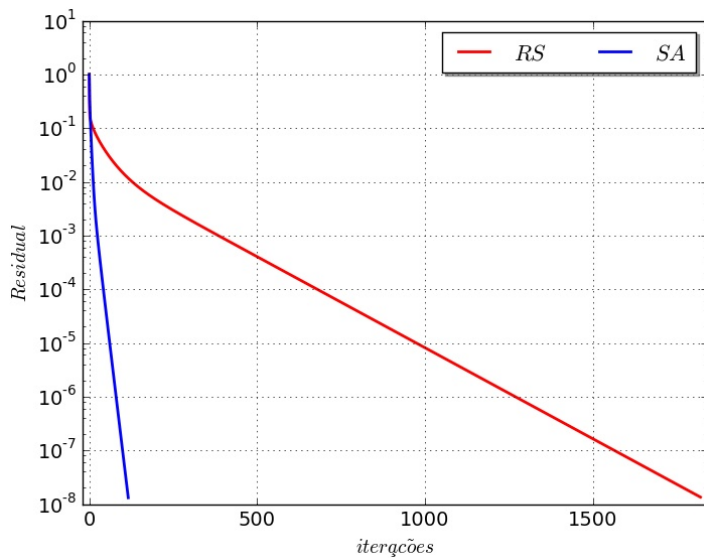


Figura 8: Residuais dos métodos iterativos: multigrid algébrico baseado em agregação suavizada, SA, e o método clássico de Ruge-Stüben, RS, para a primeira iteração do método de Picard.

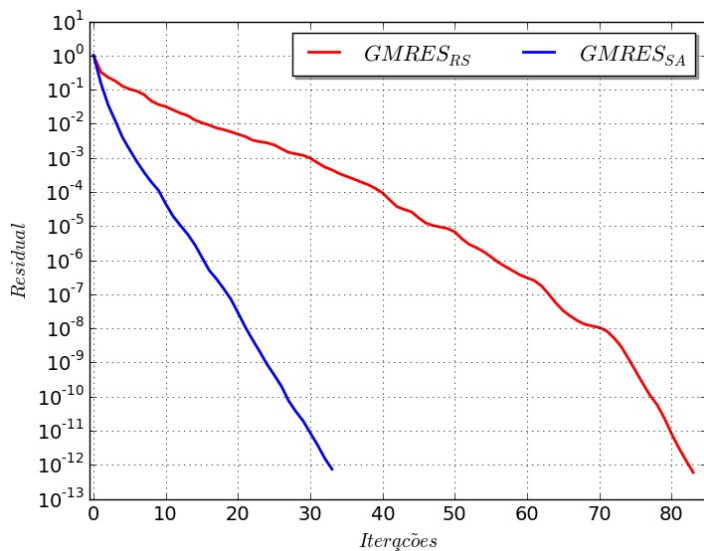


Figura 9: Residuais dos métodos iterativos $GMRES$ pré-condicionado por SA e RS para a primeira iteração do método de Picard.

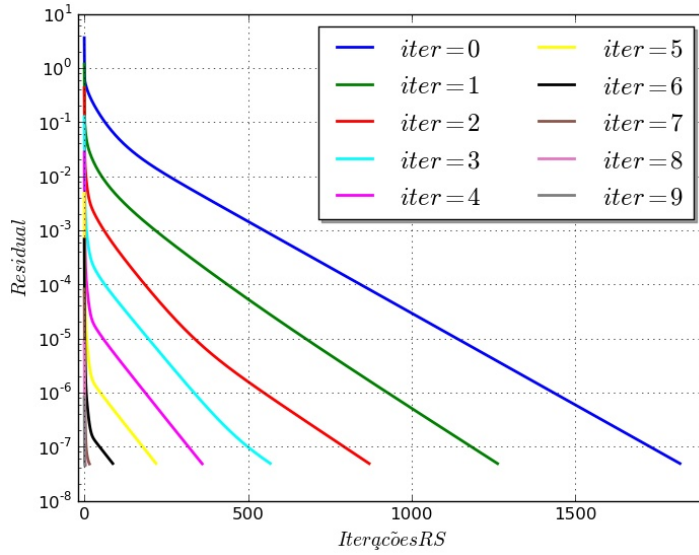


Figura 10: Residuais do multigrid algébrico clássico RS para todas as iterações de Picard.

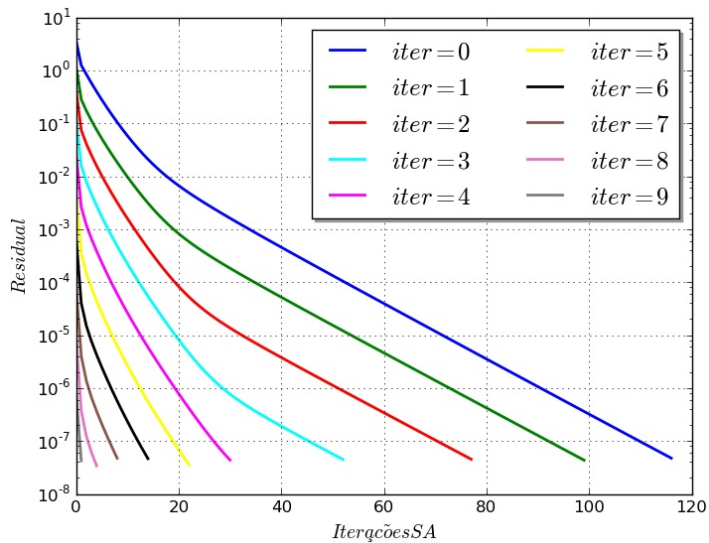


Figura 11: Residuais do multigrid algébrico baseado em agregação suavizada para todas as iterações de Picard.

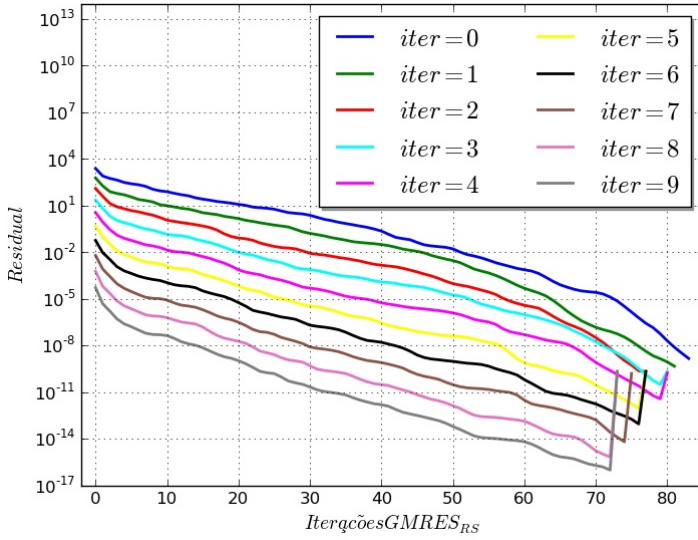


Figura 12: Residuais do método *GMRES* pré-condicionado por multigrid algébrico clássico RS, *GMRES_{RS}*, para todas as iterações de Picard.

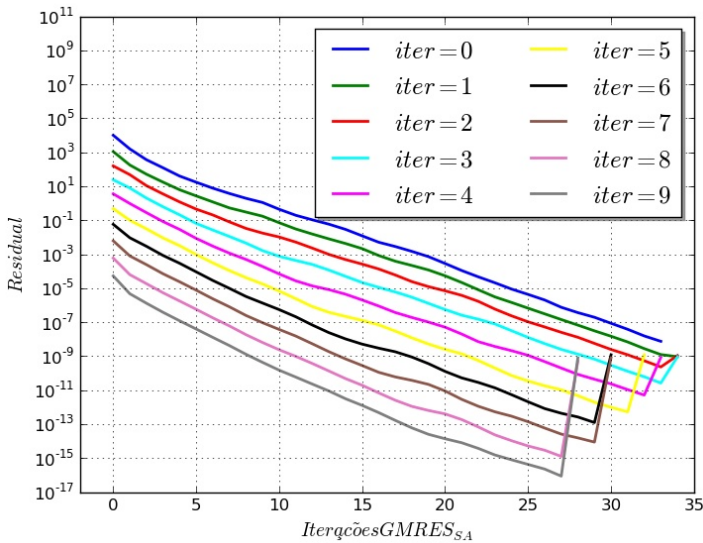


Figura 13: Residuais do método *GMRES* pré-condicionado por multigrid algébrico baseado em agregação suavizada (SA), *GMRES_{SA}*, para todas as iterações de Picard.

A análise das Figuras 12 e 13 evidenciam comportamentos mais estáveis no sentido do número de iterações. Há uma redução no número de iterações, mas não é tão expressiva quanto no caso dos métodos RS e SA. Diferentemente do caso anterior, a aplicação dos métodos $GMRES_{SA}$ e $GMRES_{RS}$ produzem soluções distintas em todas iterações de Picard, ou seja, $h_{i+1} = A^{-1} \cdot b \neq h_i, i = 0, 1, \dots, 11$ ao passo que a aplicação dos métodos RS e SA produzem soluções tais que as diferenças entre as iterações são nulas, ou seja, $h_{11} = A(h_{10})^{-1} \cdot b = h_{10}$. Portanto, ao balancear a análise anterior à análise o tempo gasto na solução dos sistemas lineares, na convergência das iterações de Picard e estabilidade do número de iterações que proporciona uma diferenciação das soluções até atingir a tolerância desejada no método de Picard, pode ser concluído que o método pré-condicionado possui melhores características no problema em questão. Ao comparar entre os métodos pré-condicionados, a tendência está em favor do método SA devido ao menor número de iterações executadas.

6 CONCLUSÕES

Os métodos multigrid algébrico de Ruge-Stüben, Agregação Suavizada e o $GMRES$ pré-condicionado por multigrid algébrico de Ruge-Stüben e agregação suavizada foram avaliados quanto ao tempo de solução dos sistemas algébricos provenientes da aproximação de elementos finitos da equação do fluxo livre estacionário. A utilização em um domínio georreferenciado apenas reforça a questão técnica e computacional, pois impõe uma dificuldade adicional para a geração das malhas, definição das condições de fronteira, etc. A análise dos métodos foi pautada pelas máximas diferenças obtidas entre as diversas soluções; pela avaliação dos residuais em cada solução do sistema linear associado à respectiva iteração de Picard; pela comparação entre o número de iterações e convergência das soluções tanto a nível do método iterativo quanto ao nível das iterações de Picard e, finalmente, pela avaliação do tempo computacional. Por fim, a análise conjunta dos elementos apresentados no artigo apontou o favorecimento dos métodos pré-condicionados em favor dos métodos não pré-condicionados ao levar em conta o tempo computacional e a estabilidade do número de iterações e consequente redução residual alcançada tanto na iteração de Picard quanto no método iterativo. Ao comparar os métodos pré-condicionados, faz-se a opção pelo método baseado em agregação.

ABSTRACT. In this paper, the algebraic multigrid method based on smoothed aggregation, the classical method of Ruge-Stüben and the method $GMRES$ preconditioned by algebraic multigrid were used to solve the steady-state free flow equation in a georeferenced domain. The availability of computer codes allowed us to evaluate the approximation of finite elements from the perspective of algebraic multigrid methods and their combination, as preconditioning, with the $GMRES$ method. The maximum differences between solutions by different methods, the time required to obtain the solutions of the associated linear systems, in each of the Picard iterations, the residuals of each of the iterative methods and the residuals of each of the Picard iterations are presented and discussed. As a result of the analysis, the preconditioned methods are more efficient in the sense of less computational time combined with the stability of the number of iterations. The analysis of the residuals of Picard iterations allows comparing the evolution of the different methods of solving linear systems. The detailing of the residuals of the iterative methods in each step of Picard's

iterations allowed a more comprehensive view and an analysis of convergence. In detail, the method based on smoothed aggregation needs a significantly smaller number of iterations when compared to the classical method of Ruge Stüben in the first iterations of Picard. The preconditioning reduces the number of iterations in relation to the initial iterations and there is a persistence of the reduction in the number of iterations of the aggregation-based method in relation to the classical method.

Keywords: finite elements, FEniCs, Python, iterative methods.

REFERÊNCIAS

- [1] R. Barrett, M.W. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine & H. van der Vorst. “Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods”. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2 ed. (1994).
- [2] J. Bear. “Hydraulics of Groundwater”. McGraw Hill, New York (2012).
- [3] K.S. ben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, (2001). doi:[https://doi.org/10.1016/S0377-0427\(00\)00516-1](https://doi.org/10.1016/S0377-0427(00)00516-1).
- [4] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge & G. Sanders. Towards adaptive smoothed aggregation (α sa) for nonsymmetric problems. *SIAM Journal on Scientific Computing*, (2010). doi:10.1137/080727336.
- [5] W.L. Briggs, V.E. Hemson & S.F. McCormick. “A Multigrid Tutorial”. Society for Industrial and Applied Mathematics, Philadelphia (1987).
- [6] J. Done & A. Huerta. “Finite Element Methods for Flow Problems”. John Wiley & Sons (2004).
- [7] A. Firmiano, J.P.M. Santos & E. Wendland. Geoprocessamento de Bacia Hidrográfica e a Solução Automatizada do Fluxo em Aquífero Freático. In “Proceeding Series of the Brazilian Society of Computational and Applied Mathematics”, volume 6 (2018).
- [8] A. Firmiano, J.P.M. Santos & E. Wendland. Geocomputação aplicada no problema de transporte acoplado ao fluxo subterrâneo em aquífero livre. *Águas Subterrâneas*, **34** (2020), 30–38. doi:<http://dx.doi.org/10.14295/ras.v34i1.29609>.
- [9] A. Firmiano, J.P.M. Santos, E. Wendland & J. Roehrig. Geoprocessamento para a Solução Fraca do Transporte de Contaminantes Acoplado ao Fluxo de Água Subterrânea. In “Trends in Applied Mathematics - TEMA”, volume 18. São Carlos (2017), p. 273–286.
- [10] M. Gockenbach. “Understanding and Implementing the Finite Element Method”. SIAM (2006).
- [11] J. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, **9** (2007), 90–95. doi:10.1109/MCSE.2007.55. URL <http://matplotlib.org/>.
- [12] L.N. L. N. Olson & J.B. J. B. Schroder. PyAMG: Algebraic Multigrid Solvers in Python, v4.0 (2018). URL <https://github.com/pyamg/pyamg>. Release 4.0.
- [13] A. Logg, K. Mardal & G. Wells. “FEniCS Project: Lecture Notes in Computational Science and Engineering”. Springer, New York (2010). doi:10.1007/978.3.642.23099.8.

- [14] J.P.M. Santos, A. Firmiano & E. Wendland. Transporte de Contaminantes em Aquífero Freático. In “Proceeding Series of the Brazilian Society of Computational and Applied Mathematics”, volume 6 (2018).
- [15] S. van der Walt, S. Colbert & G. Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*, **13** (2011), 22–30. doi:10.1109/MCSE.2011.37.
- [16] P. Vaneek, J. Mandel & M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, **56** (1995), 179–196.
- [17] R. Verfürth. Adaptive Finite Element Methods Lecture Notes Winter Term (2008).
- [18] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í. Polat, Y.F.E. W., Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt & S.. Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, **17** (2020), 261–272. doi:10.1038/s41592-019-0686-2.

