

# Uma comparação dos métodos multigrid algébricos para a solução da equação do fluxo livre em domínio georreferenciado

João Paulo Martins dos Santos<sup>1</sup>

Academia da Força Aérea - AFA

Alessandro Firmiano de Jesus<sup>2</sup>

Academia da Força Aérea - AFA

Hélio Correia Jhuniór<sup>3</sup>

Universidade de São Paulo - USP

Edson Wendland<sup>4</sup>

Universidade de São Paulo - USP

## Resumo.

Neste artigo o método multigrid algébricos baseado em agregação suavizada, o método clássico de Ruge-Stüben 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*. O tempo necessário para obter as soluções dos sistemas lineares associados às iterações de Picard, os residuais dos métodos iterativos nas respectivas iteradas de Picard e os resíduos em cada uma das iteradas de Picard são apresentados. Como resultado, o método baseado em agregação suavizada necessita de um número menor de iterações quando comparado ao método clássico de Ruge Stüben. O pré-condicionamento por método multigrid algébrico reduz drasticamente o número de iterações tanto do método *GMRES* quanto da versão pré-condicionada.

**Palavras-chave.** Elementos Finitos, FEniCs, Método de Picard, Python.

## 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, difusão e advecção, químicos e biológicos envolvidos no fluxo do fluido. Estes podem ser traduzidos matematicamente em equações diferenciais, solucionadas por métodos numéricos e 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 longitudinal e na turbulência do escoamento. Assim, a adequada determinação do campo de velocidades é essencial ao processo de

---

<sup>1</sup>jp2@alumni.br

<sup>2</sup>lezandro@gmail.com

<sup>3</sup>correiajhr@gmail.com

<sup>4</sup>ew@sc.usp.br

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, a adequada caracterização dos parâmetros de condutividade, poços de bombeamento e zonas de recarga. Finalmente, há 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 solução dos sistemas lineares associados às iterações do (das equações diferenciais 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, aos residuais dos métodos iterativos nas respectivas iteradas de Picard e o perfil dos resíduos das iterações de Picard foram obtidos para todos os métodos.

## 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 poços de bombeamento  $f$  é dada por (Ver [2]):

$$-\nabla(Kh \cdot \nabla h) + W = f, \quad \text{em } \Omega \quad (1)$$

em que  $K$  representa a condutividade hidráulica do meio e  $\Omega$  é o domínio computacional. A imposição das condições de fronteira, 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 (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 é convertido em uma sequência de sistemas lineares, denotados convenientemente por:

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

em que sucessivas aproximações  $h^1, h^2, \dots, h^{N_k}$  são 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], [18].

## 3 Método multigrid algébrico

A solução de elementos finitos de 1 é  $h = h^{N_k}$ , solução de 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].

Os métodos multigrid são classificados em geométricos ou algébricos e são diferenciados pela respectiva forma estrutural de abordagem do sistema linear. Segundo Stüben, [15], no método

geométrico a estratégia que gera a sequência de malhas é fixa e as interpolações dos resultados entre as malhas é baseada na geometria. O método algébrico, assume um método iterativo (Gauss-Seidel) e foca 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 [15] 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 [4], [15], [16], [3].

## 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 [9], os quais são complementados por [8], [7].

A implementação do método de elementos finitos considera a metodologia do Projeto FEniCS [13], a qual é baseada na utilização das formas linear e bilinear. O código computacional Python<sup>R</sup> para a equação modelo (1) e iteradas de Picard com Fatoração LU é disponibilizado em [14]. O critério de parada nas iteradas de Picard estabelecido foi o valor  $\epsilon = \max |u^{k+1} - u^k| < tol$  ou um máximo de iterações [13].

A disponibilidade dos dados e dos códigos permite 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 [12], enquanto que os métodos multigrid RS e SA são provenientes de PyAMG [5].

A visualização dos resultados utilizaram ferramenta de visualização disponível em FEniCs [13] e na biblioteca Matplotlib [11]. As manipulações das soluções foi realizada com biblioteca numérica NumPy [17].

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 [9]. 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 que uma estimativa aleatória ou nula.

## 5 Resultados

Inicialmente o problema do fluxo livre e estacionário foi resolvido com o método SA e com o método RS. Em seguida os métodos  $GMRES_{RS}$   $GMRES_{SA}$  foram utilizados. O código foi modificado de modo a inserir um termo de recarga adicional 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 1. As curvas de nível de  $h$  sobrepostas ao campo de velocidades normalizado são apresentados na Figura 2. Ambas contém informações que favorecem a análise qualitativa do comportamento de um contaminante presente no meio poroso, analisar a influência do conjunto de poços ou zonas de recarga sobre o fluxo subterrâneo, ou, além disso, analisar a influência de um elementos específico sobre o domínio. Ressalta-se que estes resultados diferem daqueles apresentados em [9] devido à presença adicional de um termo de recarga.

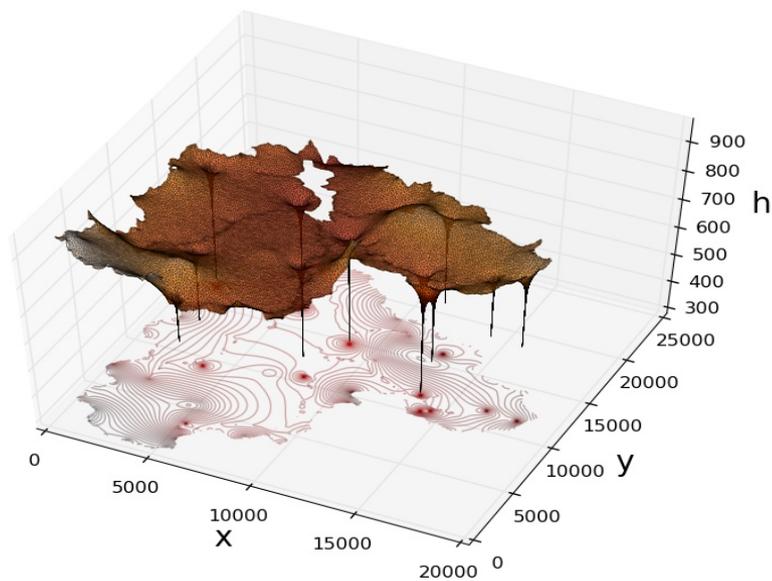


Figura 1: Solução numérica da equação do fluxo em domínio georreferenciado. Fonte: Modificado de [9] para englobar um termo de recarga adicional

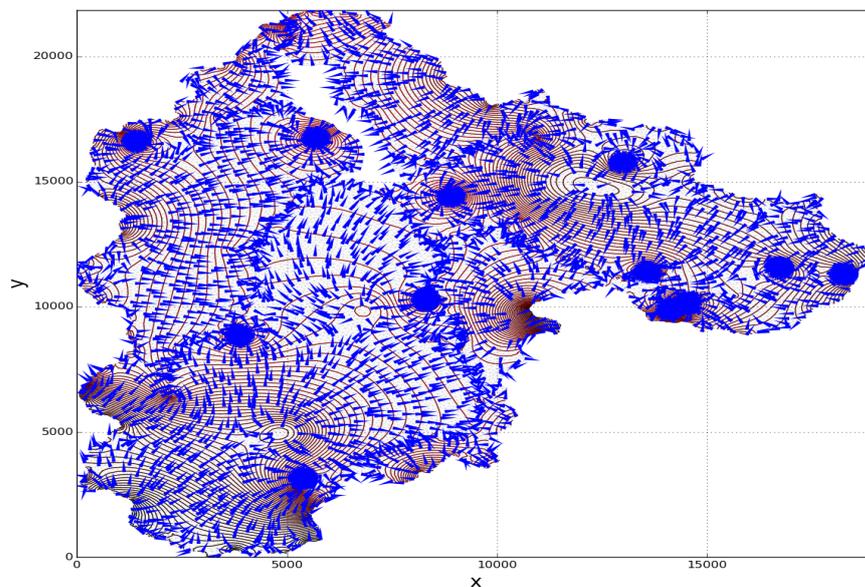
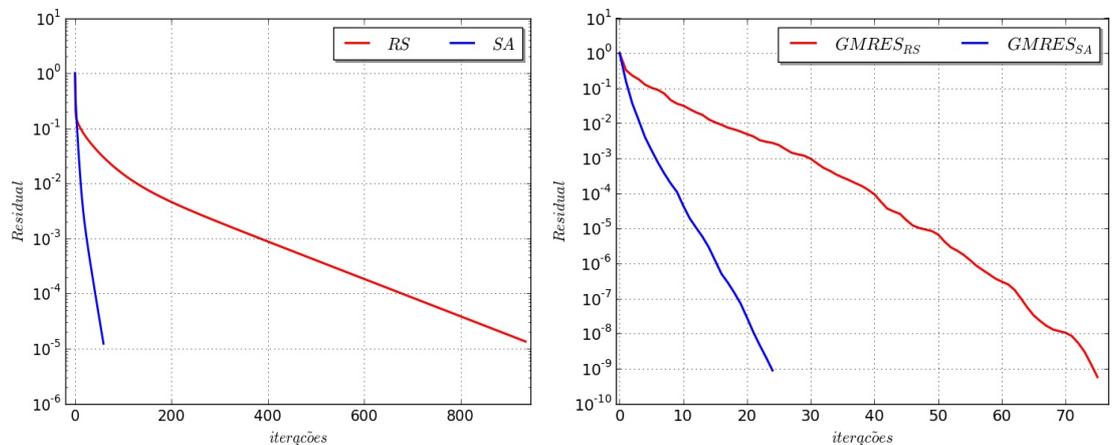


Figura 2: Campo de velocidades e curvas de nível da solução numérica da equação do fluxo em domínio georreferenciado. Fonte: Modificado de [9] para englobar um termo de recarga adicional

As Figuras 3-a, b mostram os residuais na primeira iteração de Picard. A primeira mostra os residuais para os métodos  $RS$  e  $SA$ , enquanto que a última mostra os resultados dos métodos  $GMRES_{RS}$  e  $GMRES_{SA}$ .

Ao analisar a Figura 3-a, é notável que o número de iterações necessárias para a convergência do método *SA* é muito menor que aquele requerido pelo método *RS*. Neste caso, ambos os parâmetros foram escolhidos arbitrariamente e sem qualquer tipo de procedimento para otimização. Ressalta-se que ambos os métodos são influenciados pela escolha dos parâmetros e, portanto, configurações distintas podem afetar as respectivas performances. No entanto, após algumas configurações distintas para o método *RS*, os resultados ainda mostravam o mesmo comportamento qualitativo do método *RS* em relação ao método *SA*.

De forma análoga, a Figura 3-b mostra os residuais do método  $GMRES_{RS}$  e  $GMRES_{SA}$  na primeira iteração de Picard. Novamente, o número de iterações associado ao método *SA* é menor que aquele associado ao método *RS*, mas com um significativo aumento na proporção dos dois números. Ao comparar o método  $GMRES_{SA}$  e  $GMRES_{RS}$  com os respectivos métodos *SA* e *RS*, nota-se uma redução drástica do número de iterações juntamente com uma acentuada redução dos respectivos residuais. A redução do número de iterações dos métodos pré-condicionados é ainda mais drástica quando comparado ao método *GMRES*, o qual converge em, aproximadamente, 4000 iterações para os parâmetros estabelecidos no pacote *linalg*, disponível em SciPy [12].



(a) Residuais dos métodos *SA* e *RS* para primeira iteração do método de Picard. (b) Residuais dos métodos  $GMRES_{RS}$  e  $GMRES_{SA}$  para primeira iteração do método de Picard.

Figura 3: Residuais dos métodos iterativos *SA*, *RS*, *GMRES* pré-condicionado por *SA* e *RS* para a primeira iteração do método de Picard.

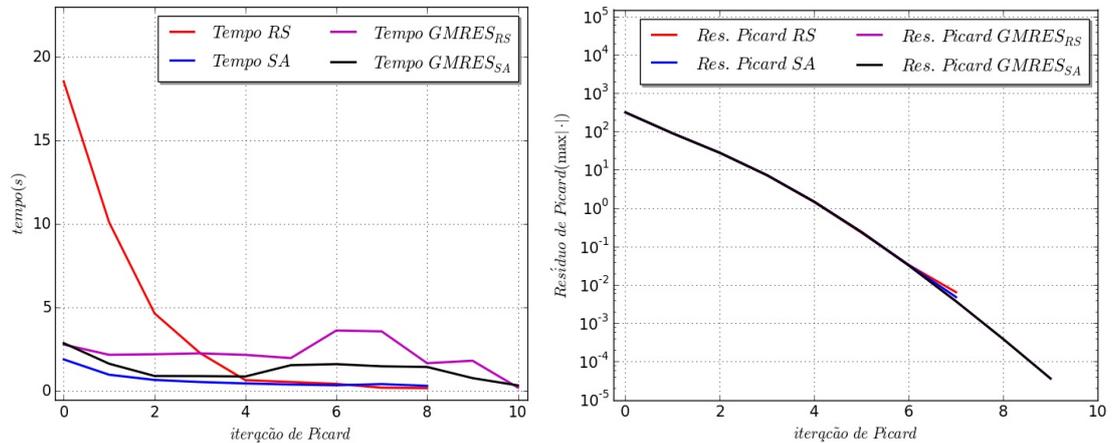
As demais iterações de Picard não foram analisadas quanto à evolução dos residuais conforme apresentado nas Figuras 3-a,b. No entanto, os resultados seguintes contemplam implicitamente o número de iterações necessárias em cada passo do método iterativo de Picard.

As Figuras 4-a,b mostram, respectivamente, os tempos necessários para calcular a solução em cada passo do método de Picard e as respectivas evoluções do algoritmo, de acordo com a norma do máximo, até o passo anterior à convergência.

Quanto ao tempo, a análise dos resultados mostra que o método *RS* requer, aproximadamente, quatro vezes mais tempo no início do processo de Picard. Apesar do comportamento decrescente, o menor tempo é equivalente ao tempo dos métodos *SA* e  $GMRES_{SA}$ . Os métodos *SA* e  $GMRES_{SA}$ , por sua vez, possuem tempos aproximadamente iguais no início do processo, com uma divergência das diferenças a partir da quinta iteração de Picard. Finalmente, o método  $GMRES_{RS}$  segue o mesmo padrão que o método  $GMRES_{SA}$ , mas com um tempo ligeiramente maior.

Da análise dos resultados da Figura 4-b, conclui-se que os métodos pré-condicionados requerem iterações extras. Os valores são visualmente idênticos nas primeiras iterações, mas há uma aparente

divergência a partir da sétima iteração. Ao utilizar os métodos *SA* e *RS*, o processo de Picard converge na oitava iteração, enquanto dez iterações são necessárias ao utilizar os métodos pré-condicionados. Ressalta-se que uma tolerância de  $\varepsilon = 10^{-6}$  e um número máximo de 400 iterações foram utilizadas em todos os casos e todos os métodos utilizaram como critério de parada os mesmos valores para a tolerância e número de iterações ( $tol = 10^{-9}$  e  $MaxIter = 10000$ ).



(a) Tempo requerido em cada iteração do método de Picard. (b) Resíduos em cada uma das iterações do método de Picard.

Figura 4: *SA*, *RS*, *GMRES* pré-condicionado por *SA* e *RS* para a primeira iteração do método de Picard.

## 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 do número de iterações necessárias à convergência, a avaliação do tempo necessário em cada iteração e a análise dos resíduos do processo iterativo de Picard, permite estabelecer uma ordenação entre os métodos avaliados e, além disso, evidenciar a influência da pré-condicionamento por método multigrid algébrico.

## Referências

- [1] Barrett, R. and Berry, M.W. and Chan, T.F. and Demmel, J. and Donato, J. and Dongarra, J. and Eijkhout, V. and Pozo, R. and Romine, C. and van der Vorst, H., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 1994.
- [2] Bear, J. *Hydraulics of Groundwater*, New York, McGraw Hill, 2012.
- [3] Brezina, M. and Manteuffel, T. and McCormick, S., Ruge, J. and Sanders, G. Towards adaptive smoothed aggregation (asa) for nonsymmetric problems. *SIAM Journal on Scientific Computing*, 2010. DOI: 10.1137/080727336.

- [4] Briggs, W. L., Hemson, V.E., McCormick, S.F., *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 1987.
- [5] Bell, W. N. and Olson, L. N. and Schroder, J., *PyAMG: Algebraic Multigrid Solvers in Python*, 2013, Disponível em: <http://www.pyamg.org>.
- [6] Donea, J. and Huerta, A. *Finite Element Methods for Flow Problems*, John Wiley & Sons, 2004.
- [7] Firmiano, A., Santos, J.P.M., Wendland, E., Roehrig, J. Geoprocessamento para a Solução Fraca do Transporte de Contaminantes Acoplado ao Fluxo de Água Subterrânea. *Trends in Applied Mathematics -TEMA* , São Carlos , v. 18, n. 2, p. 273-286, Aug. 2017.
- [8] Firmiano, A., Santos, J.P.M., Wendland, E., 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, vol. 6, n 1, 2018.
- [9] Firmiano, A., Santos, J.P.M., Wendland, E., Geocomputação aplicada no problema de transporte acoplado ao fluxo subterrâneo em aquífero livre, *Águas Subterrâneas*, v. 34, n. 1, p. 30-38, 2020. DOI: <http://dx.doi.org/10.14295/ras.v34i1.29609>
- [10] Gockenbach, M. *Understanding and Implementing the Finite Element Method*, SIAM, 2006.
- [11] Hunter, J.D. Matplotlib: A 2D graphics environment, *Computing In Science & Engineering*, v.9, n.3, pg. 90-95, 2007. DOI = 10.1109/MCSE.2007.55—. Disponível em <http://matplotlib.org/>.
- [12] Jones, E. et al. SciPy: Open source scientific tools for Python, 2001. Disponível em: <http://www.scipy.org/>.
- [13] Logg, A. and Mardal, K.A. and Wells, G. *FEniCS Project: Lecture Notes in Computational Science and Engineering*, Springer, New York, 2010. DOI 10.1007/978.3.642.23099.8
- [14] Santos, J.P.M. e Firmiano, A. e Wendland, E., Transporte de Contaminantes em Aquífero Freático, In: Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, vol 6, n 1, 2018.
- [15] Stüben, K., 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).
- [16] Vanek, P. and Mandel, J. and Brezina, M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing* 1995; **56**:179–196.
- [17] van der Walt, S. and Colbert, S.C. and Varoquaux, G. *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science Engineering, 2011. vol 13, n 2.
- [18] Verfürth, R. *Adaptive Finite Element Methods Lecture Notes Winter Term*, Fakultät für Mathematik, Ruhr-Universität Bochum, Deutschland, 2008.