

Um Método Newton-Inexato com Estratégia Híbrida para Globalização¹

R.G. BEGIATO², M.A. GOMES-RUGGIERO³, Departamento de Matemática Aplicada, IMECC, UNICAMP, 13083-970 Campinas, SP, Brasil.

Resumo. Neste trabalho, o objetivo é propor um algoritmo Newton-inexato com propriedade de convergência global para resolução de sistemas não-lineares. Para a globalização, propomos uma abordagem híbrida, envolvendo, além de busca linear, o método de regiões de confiança *Dogleg*. Para a resolução dos sistemas lineares, optamos por usar o método GMRES, que permite o uso implícito das matrizes e possibilita trabalhar com a estratégia *matrix-free*.

Palavras-chave. sistemas não-lineares, estratégia de globalização, GMRES(m).

1. Introdução

A proposta deste trabalho é a resolução de sistemas não-lineares

$$F(x) = 0, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n \text{ e } F \in C^1(\mathbb{R}^n). \quad (1.1)$$

Entre os métodos clássicos para a resolução de sistemas não-lineares encontra-se o método de Newton. Denotando a matriz Jacobiana de F por J , o método de Newton aplicado à resolução de (1.1) consiste em realizar iterações do tipo, $x_{k+1} = x_k + s_k$, sendo x_k o ponto atual e s_k a direção obtida pela solução do seguinte sistema linear

$$J(x_k)s = -F(x_k). \quad (1.2)$$

O método de Newton tem como principal vantagem a taxa de convergência quadrática, no entanto apresenta algumas dificuldades na resolução de problemas de grande porte, pois o cálculo da matriz Jacobiana, bem como a posterior resolução do sistema linear podem tornar-se computacionalmente caros ou até impraticáveis.

Dembo, Eisenstat e Steihaug propuseram, em 1982, o método Newton-Inexato [4], que não preserva a taxa de convergência quadrática do método de Newton, mas requer um trabalho computacional consideravelmente menor por iteração. A proposta do método é: a cada iteração k , dado o parâmetro $\eta_k \in (0, 1]$, obter uma direção s_k satisfazendo

$$\|J(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\|. \quad (1.3)$$

¹trabalho realizado com auxílio do CNPq.

²rodolfo@ime.unicamp.br

³marcia@ime.unicamp.br

As variações entre as implementações do método Newton-inexato ocorrem basicamente por causa de diferentes escolhas que pode-se adotar em três etapas do método:

1. método para resolução do sistema linear (1.2);
2. escolha para os valores do parâmetro η_k ;
3. estratégia para garantir a convergência global do Newton-inexato.

A seguir descreveremos as escolhas em cada uma das etapas acima e sobre as implementações e testes realizados.

2. Resolução do Sistema Linear

O passo s_k que satisfaz a condição (1.3) pode ser obtido através de um método iterativo para resolução de sistemas lineares. Neste trabalho optamos por empregar o método GMRES (*Generalized Minimal Residual*) [11], que é um método da família de métodos de projeção sobre subespaços de Krylov.

Dado um sistema linear $As = b$ e definindo $r_0 = b - As_0$ o resíduo inicial, a p -ésima iteração do GMRES consiste em obter a solução do problema de minimização

$$\begin{aligned} \min \|b - As\|_2 \\ \text{s. a: } s \in s_0 + \mathcal{K}_p(A, r_0), \end{aligned} \quad (2.1)$$

onde $\mathcal{K}_p(A, r_0)$ denota o subespaço de Krylov de ordem p , gerado por: $r_0, Ar_0, A^2r_0, \dots, A^{p-1}r_0$. O algoritmo para o GMRES inclui um processo de ortogonalização dos vetores geradores do subespaço de Krylov, que é realizado através do processo de Arnoldi. Ao final de cada iteração deste procedimento teremos a matriz V_p , cujas colunas são os p vetores ortonormalizados; a matriz V_{p+1} que é a matriz V_p acrescida em um vetor coluna e a matriz Hassenberg superior \bar{H}_p . Tais matrizes satisfazem a relação: $AV_p = V_{p+1}\bar{H}_p$. Dado que $s \in s_0 + \mathcal{K}_p(A, r_0)$, existe $y \in \mathbb{R}^p$ de tal forma que, podemos escrever o vetor s como $s = s_0 + V_p y$. Denotando $\beta = \|r_0\|_2$, e considerando que $v_1 = r_0/\|r_0\|_2$, podemos reescrever o problema (2.1), e minimizar irrestritamente $\|\beta e_1 - \bar{H}_p y\|_2$.

Considerando que a matriz \bar{H}_p é Hessemberg superior, conseguimos facilmente encontrar sua fatoração QR : $\bar{H}_p = Q_p R_p$. Assim, o problema (2.1) torna-se um problema de fácil resolução, pois é equivalente a minimizar $\|\beta Q_p^t e_1 - R_p y\|_2$ com $y \in \mathbb{R}^p$.

Se na j -ésima iteração do GMRES tivermos $A^j r_0 = 0$, o método GMRES não poderá prosseguir. Resultados garantem que esta situação ocorre, se somente se, s_j é a solução exata do sistema $As = b$ (ver [9]).

Devido à sua estrutura, o GMRES exige o uso da matriz Jacobiana somente no produto entre esta e outro vetor v , o que permite que trabalhem com estratégia *matriz-free*, isto é, aproximamos o resultado dessa operação por meio da seguinte expansão por série de Taylor

$$J(x_k)v \approx (1/h)(F(x_k + hv) - F(x_k)), \quad h \in \mathbb{R}. \quad (2.2)$$

Embora o GMRES apresente várias facilidades de implementação, seu custo computacional e os requerimentos de memória aumentam conforme o número de

iterações realizadas. Por esta razão, é usual implementar o método GMRES com recomeços a cada m iterações, GMRES(m), que consiste em realizar no máximo m iterações do GMRES e caso não haja convergência recomeçamos o processo, utilizando o valor encontrado na última iteração como aproximação inicial. Embora essa estratégia não tenha resultados teóricos tão satisfatórios quanto aqueles obtidos para o caso sem recomeços, o uso prático desta abordagem é necessário para problemas de grande porte e tem apresentado bons resultados.

3. Escolha para os Parâmetros η_k

A escolha dos valores para o parâmetro η_k é de grande importância para o desempenho do método, uma vez que tem influência direta em sua taxa de convergência. Um valor muito pequeno certamente possibilitará a realização de poucas iterações do tipo $x_{k+1} = x_k + s_k$, as quais chamaremos de iterações externas. Porém, valores muito próximos de zero provavelmente podem resultar no fenômeno conhecido na literatura como *oversolving*, que consiste na realização de um grande número de iterações do GMRES (iterações internas). A taxa superlinear de convergência é obtida se a sequência de valores η_k é tal que $\lim_{k \rightarrow \infty} \eta_k \rightarrow 0$, [4]. Neste trabalho optamos pela fórmula apresentada por Eisenstat e Walker em [8]

$$\eta_k = \gamma (\|F(x_k)\|_2 / \|F(x_{k-1})\|_2)^\alpha, \quad (3.1)$$

que resulta num algoritmo com taxa superlinear de convergência.

4. Estratégia para Globalização

Assim como os métodos de Newton, o método Newton-Inexato tem sua convergência assegurada somente nas regiões próximas à solução. Para obter resultados de convergência global é necessário adotar uma estratégia para globalização, que por sua vez são associadas a problemas de minimização: $\min f(x)$, $x \in \mathbb{R}^n$. Duas propostas amplamente usadas são busca linear e regiões de confiança.

Quando usamos a estratégia de busca linear, devemos inicialmente encontrar uma direção s_k e em seguida, um tamanho de passo que cumpra alguma condição de decréscimo no valor da função f . No caso de utilizarmos regiões de confiança, aproximamos f , em torno de x_k , por um modelo quadrático e fixamos o raio da região, δ_k , que supomos poder confiar que o modelo aproxima adequadamente a função f . E então, a próxima direção, s_k , será dada pelo minimizador do modelo quadrático na região determinada pelo raio da região de confiança δ_k . Em outras palavras devemos resolver o seguinte problema de minimização

$$\begin{aligned} \min & f(x_k) + \nabla f(x_k)^t s + \frac{1}{2} s^t H_k s \\ \text{s.a:} & \|s\|_2 \leq \delta_k. \end{aligned} \quad (4.1)$$

O Lema 4.1 apresenta uma solução para o problema acima, e é fundamental na construção de nosso algoritmo. A Figura 1(a), expressa o resultado do Lema. Sua demonstração pode ser encontrada em [6].

Lema 4.1. *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R} \in C^2$, $H_k \in \mathbb{R}^{n \times n}$ simétrica e definida positiva. Então,*

(i) *o Problema (4.1) é resolvido por $s(\mu) \triangleq -(H_k + \mu I)^{-1} \nabla f(x_k)$ com o único $\mu \geq 0$ tal que $\|s(\mu)\|_2 = \delta_k$, a menos que $\|s(0)\|_2 \leq \delta_k$, e neste caso $s(0) = s_k^N$ (o passo completo de Newton) é a solução;*

(ii) *para qualquer $\mu \geq 0$, $s(\mu)$ define uma direção de descida de f em x_k .*

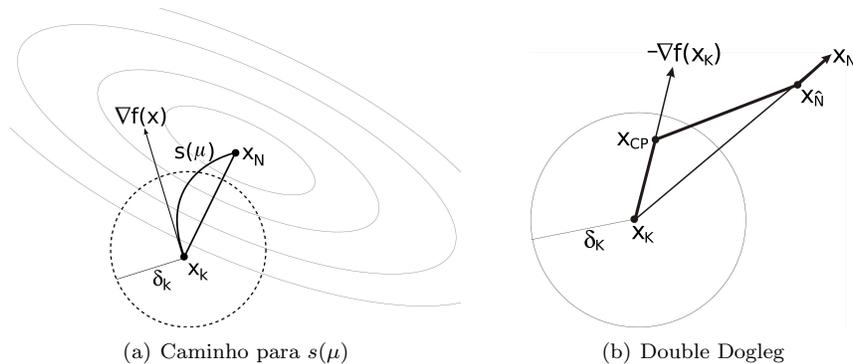


Figura 1: Aproximação *Double Dogleg*

Seguindo as idéias propostas por Brown e Saad em [3], utilizamos a estratégia de regiões de confiança conhecida como *Double Dogleg* proposta por Dennis e Mei em 1979 [5]. Essa estratégia consiste em aproximar a curva $s(\mu)$ através de uma curva linear por partes. Para isso devemos encontrar a direção de máxima descida (s_M) e a direção de Newton (s_N), ambas para o modelo quadrático do problema (4.1). Obtemos então dois pontos: o minimizador do modelo quadrático na direção de descida, chamado ponto de Cauchy e denotado por x_{CP} e o ponto $x_{\hat{N}} = x_k + \nu s_N$, onde $\nu \in (0, 1)$ e deve ser escolhido de maneira conveniente a tornar o algoritmo bem definido.

Definimos o caminho $x_k \rightarrow x_{CP} \rightarrow x_{\hat{N}} \rightarrow x_N = x_k + s_N$ e, caso o ponto encontrado pelo passo completo de Newton se localize fora da região de confiança, definimos como próximo passo o ponto na intersecção entre a fronteira da região de confiança e esse caminho (ver Figura 1(b)).

Utilizamos $f(x) = \|F(x)\|_2^2$ como função de mérito. Uma vez que a direção s_k resultante do GMRES com recomeços é escrita na forma $s_k = s_0 + V_p y$, podemos reescrever o problema $\min f(x)$, $x \in \mathbb{R}^n$, na variável y : $\min g(y) = f(x_k + s_0 + V_p y)$, $y \in \mathbb{R}^p$. Dado que empregamos o modelo linear $F + Js$ para aproximar $F(x_k + s)$, temos que uma aproximação quadrática natural para $g(y)$ é a função $\frac{1}{2} \|F + Js\|_2^2$. Definindo $\bar{y} = [y, t]^t$ e $W = [V_p, s_0]$, o modelo quadrático é dado por $\hat{g}(\bar{y}) = \frac{1}{2} \|F + JW\bar{y}\|_2^2$. Encontramos, então, a direção de máxima descida para o modelo quadrático

$$s_M = -\nabla \hat{g}(0) = -(JW)^t F = -[JV_p \quad Js_0]^t F = [-F^t V_{p+1} \bar{H}_p \quad (F + r_0)^t F]^t.$$

Para encontrar a direção de Newton, (s_N) , observamos inicialmente que: $Bs_N = s_M$, onde $B = (JW)^t(JW)$ é a Hessiana do modelo quadrático. Considerando então $v = -\bar{H}_p^t V_{p+1}^t (F + r_0)$ e $a = \|F + r_0\|_2^2$ podemos reescrever: $B = (JW)^t(JW) = \begin{pmatrix} \bar{H}_p^t \bar{H}_p & v \\ v^t & a \end{pmatrix}$. Durante a implementação é conveniente lembrar que o método GMRES obteve, a fatoração QR da matriz \bar{H}_p . Logo $\bar{H}_p^t \bar{H}_p = (QR)^t(QR) = R^t R$. Como a dimensão de \bar{H}_p é $(p+1) \times p$, a matriz $R : (p+1) \times p$ terá a seguinte estrutura: $R = [\bar{R}, 0]^t$ onde $\bar{R} \in \mathbb{R}^{p \times p}$. Considerando $w = \bar{R}^{-1}v$ e $b = \sqrt{a - w^t w}$, obtemos a decomposição de Cholesky para B

$$B = \begin{pmatrix} \bar{R}^t & 0 \\ w^t & b \end{pmatrix} \begin{pmatrix} \bar{R} & w \\ 0 & b \end{pmatrix} \equiv C^t C.$$

Pela estratégia de globalização, devemos aceitar pontos que satisfaçam um decréscimo suficiente na função f . Embora o mais comum seja adotar uma estratégia que exige decréscimo monótono, alguns autores têm trabalhado com a estratégia de decréscimo não-monótono. Optamos por esta estratégia e utilizamos a técnica proposta por Birgin, Krejić e Martínez em [2], que consiste em considerar o ponto x_k satisfatório se cumprir a condição:

$$\|F(x_k + \xi s_k)\| < (1 - \xi\sigma)\|F(x_k)\| + \mu_k, \quad \sigma \in (0, 1), \quad 0 < \xi \leq 1, \quad \sum_{k=0}^{\infty} \mu_k < \infty. \quad (4.2)$$

5. Implementação e Testes Realizados

As estratégias de globalização que incorporamos ao algoritmo Newton-inexato são híbridas, no sentido que numa primeira fase trabalham com procedimento de busca linear ao longo da direção s_k obtida pelo método GMRES. O ponto é aceito se satisfaz o teste (4.2), caso contrário, o tamanho do passo é reduzido. Este processo é realizado no máximo três vezes. Se nesta fase não for obtido um ponto que satisfaça o decréscimo monótono, a estratégia de regiões de confiança é acionada.

O primeiro algoritmo implementado, ao qual chamaremos **3BL1**, consiste na utilização do critério (4.2) tanto para aceitação de pontos encontrados pela estratégia de busca linear quanto aqueles encontrados pela estratégia de região de confiança, *Double Dogleg*.

O segundo algoritmo **3BL2**, difere do primeiro somente no critério de aceitação de pontos advindos da estratégia *Double Dogleg*. Neste caso, ao invés de utilizarmos o critério (4.2), utilizamos $|\Delta f_{Pred} - \Delta f_{Ared}| \leq 0.1|\Delta f_{Ared}|$, aqui, Δf_{Ared} representa a redução real e Δf_{Pred} a redução prevista em f .

Feito isso, comparamos o desempenho entre esses algoritmos e os algoritmos sem busca linear (**SBL**) e sem globalização (**SGLOB**). Resolvemos 16 problemas acadêmicos, propostos em [10], todos com dimensão $n = 1000$. Cada problema foi resolvido com 10 aproximações iniciais: $(0, \dots, 0)$, x_{ones} , $2x_{ones}$, $5x_{ones}$, x_{stand} , $2x_{stand}$, $5x_{stand}$, $-x_{stand}$, $-2x_{stand}$ e $-5x_{stand}$, onde x_{stand} é o valor inicial padrão proposto em [10] e $x_{ones} = (1, \dots, 1)$.

Na sequência de termos forçantes (3.1) utilizamos os parâmetros: $\alpha = 0.5(1 + \sqrt{5})$ e $\gamma = 1$. Além disso, foi introduzida uma salvaguarda de modo que $\eta_k \in [10^{-6}, 10^{-2}]$. Para o critério (4.2), escolhemos o parâmetro $\sigma = 10^{-4}$. Inicializamos o parâmetro ξ com $\xi = 1$, e enquanto o critério de aceitação não se verifica, definimos $\xi = 0.5\xi$. O parâmetro μ_k é obtido de acordo com a expressão $\mu_k = ftip(k)/(k+1)^{1.1}$, onde $ftip(k)$ é o menor valor para $\|F\|_2$ atualizado somente a cada três iterações.

Estabelecemos como critérios de parada: $\|F(x_k)\|_2 \leq \sqrt{n}10^{-6}$ ou o número máximo de iterações k , $k > 100$. Utilizamos $m = 30$ como o número máximo de iterações para cada ciclo do GMRES e estabelecemos o limite máximo de 20 ciclos. Para o valor de h na implementação *matrix-free* (2.2), testamos a proposta de Dennis e Schnabel (**h:DS**): $h = (1/\|v\|_2)\sqrt{\varepsilon} \max\{|x_k^t v|, tipx_k^t|v|\} \text{sign}(x_k^t v)$, onde ε é o erro da máquina, $tipx_k$ é um vetor composto por valores típicos para x_k e $\text{sign}(u)$ é uma função que retorna 1 caso u seja positivo e -1 caso u seja negativo.

Para comparar o desempenho dos algoritmos, usamos uma ferramenta chamada *análise de desempenho*, desenvolvida por Dolan e Moré ([7]). Esta ferramenta consiste em plotar um gráfico cumulativo de forma que podemos comparar o desempenho dos algoritmos tanto quanto à eficiência (maiores valores para $\tau = 1$ indicam algoritmos mais eficientes), quanto à robustez (algoritmos que atingirem $\rho(\tau) = 1$ primeiro são considerados mais robustos, lembrando que se o algoritmo não converge para algum problema, ele atingirá $\rho(\tau) = 1$ em um valor bem alto para τ). Segue na Figura 2 o gráfico de desempenho dos algoritmos em número de iterações externas e internas. Ressaltamos que todos os testes computacionais foram realizados em Matlab® 7.0.

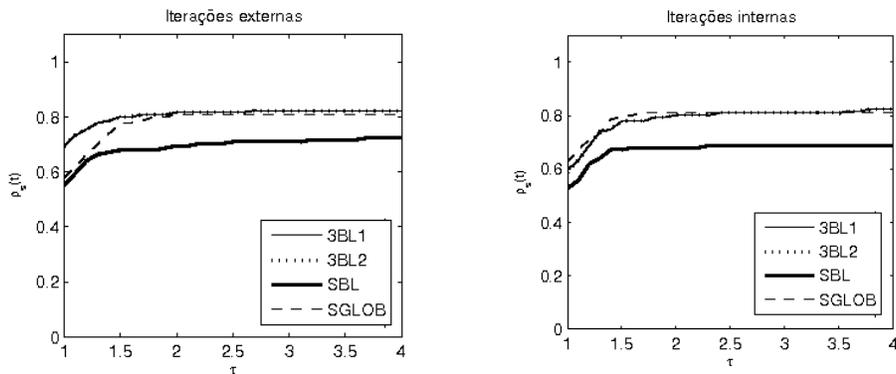


Figura 2: Desempenho nos testes de [10]

Observamos que os algoritmos com estratégia híbrida obtiveram desempenho ligeiramente superior que o algoritmo sem globalização, o que nos leva a concluir que a estratégia de globalização foi pouco acionada em problemas que convergiram. Além disso, é possível verificar o desempenho inferior do algoritmo sem busca linear, esse algoritmo é mais rígido que os demais, o que pode não ter permitido um avanço maior quando estávamos distantes da solução.

Comparamos o desempenho do algoritmo **3BL1**, com o mesmo algoritmo, mas usando a proposta de Bellavia e Morini (**h:BM**) para o valor de h na estratégia *matrix-free*: $h = \sqrt{\varepsilon} \|x_k\|_2 / \|v\|_2$, aqui ε é novamente a precisão da máquina. O algoritmo, com essa nova escolha para o valor de h obteve desempenho superior em todas as medidas de comparação escolhidas (iterações externas e internas, tempo de execução e número de avaliações de função). Finalizando esta bateria de testes, comparamos o desempenho entre o algoritmo **3BL1**, com escolha do parâmetro **h:BM** e o algoritmo **ABM** proposto por Bellavia e Morini em [1], no problema (*Extended Powell Badly Scaled*) constante em [10], para $n = 4096$ e os mesmos valores iniciais. Os resultados encontram-se na Tabela 1, onde estão denotados por * os testes que não apresentaram convergência. Devido à estrutura do algoritmo

Tabela 1: Resultados para o problema *Extended Powell Badly Scaled*([10])

x_0	Iter. Externas		No. Aval. de F	
	ABM	3BL1	ABM	3BL1
$(0, \dots, 0)$	*	33	*	360
x_{ones}	133	17	1040	167
$2x_{ones}$	134	16	1031	152
$5x_{ones}$	125	26	918	274
x_{stand}	135	17	1034	164
$2x_{stand}$	133	16	1029	154
$5x_{stand}$	125	26	1009	272
$-x_{stand}$	*	15	*	151
$-2x_{stand}$	*	29	*	301
$-5x_{stand}$	*	29	*	290

ABM, já esperava-se um número menor de avaliações de função no algoritmo **3BL1**, porém a eficiência em termos de número de iterações externas, bem como a robustez de **3BL1**, foi surpreendente.

No segundo conjunto de testes, resolvemos uma série de problemas não-lineares oriundos da discretização de problemas de valor de contorno

$$-\nabla u + h(\lambda, u) = w(s, t) \text{ em } \Omega, \quad u(s, t) = 0 \text{ em } \partial\Omega, \quad (5.1)$$

onde ∇u é o operador Laplaciano. Os diferentes problemas de valor de contorno são definidos através de variações na função $h(\lambda, \mu)$, do parâmetro λ e da função w . Seguindo as idéias encontradas em [9], utilizamos o método de diferenças centrais para aproximar as derivadas. Trabalhamos com uma malha de 63 pontos internos em cada eixo, e do processo de discretização, a resolução numérica do problema (5.1) resultou em um sistema não linear de dimensão 3969.

Resolvemos dois problemas desse tipo: o problema de Bratu, onde temos que $h(\lambda, u) = -\lambda \exp(u)$ na equação (5.1) e o problema de convecção-difusão, onde teremos $h(\lambda, u) = \lambda u \left(\frac{\partial u}{\partial s} + \frac{\partial u}{\partial s} \right)$.

A idéia é comparar os resultados encontrados com a solução real e para isto, conforme sugerido em [9], geramos uma série de problemas onde a função w é

calculada de tal forma que a solução exata do sistema resultante seja dada por: $u_* = 10st(1-s)(1-t)\exp(s^{4.5})$.

Para o problema de Bratu geramos 22 testes com 11 problemas diferentes (os valores para λ escolhidos foram: $-1000, -500, -250, -100, -50, -10, 1, 3, 5, 7, 10$) e duas aproximações iniciais ($x_0 = (0, \dots, 0)$ e x_0 com entradas aleatórias no intervalo $[-5, 5]$). No caso do problema de convecção-difusão optamos por λ : 5, 10, 25, 50, 75, 100, 110, 125 e 150 e para a aproximação inicial utilizamos o vetor $x_0 = (0, 0, \dots, 0)^t$.

O desempenho dos algoritmos nos testes para a resolução do problema de Bratu, foi semelhante e houve convergência para todos os problemas, com todos os algoritmos, sendo que a estratégia de globalização foi pouco ou nunca acionada, e por esta razão, omitimos as tabelas comparativas.

Para o problema de convecção-difusão, o algoritmo sem globalização obteve resultado inferior aos demais algoritmos, já que divergiu em 4 problemas, enquanto os demais divergiram em apenas um problema conforme pode ser conferido pela Tabela 2. Nesta tabela está representado o desempenho nos problemas considerados mais difíceis, **EX** representa o número de iterações externas, **IN** o número de iterações internas e $e = \|x^* - x\|_\infty$, onde x^* é a solução exata para o sistema linear e x é a solução encontrada pelo algoritmo em questão. Os casos em que $e < 10^{-8}$ estão denotados por —. O único algoritmo testado com a proposta de Bellavia e Morini para o parâmetro h na estratégia *matrix-free* foi **3BL1**, por isso fizemos a distinção na tabela.

Tabela 2: Resultados para o problema de Convecção-Difusão

		Teste para $m = 30$				Teste para $m = 50$				
		λ	100	110	125	150	100	110	125	150
3BL1 h: BM	e	—	—	—	0.7350	—	—	—	—	—
	EX	18	21	26	100	18	21	26	34	34
	IN	3693	4840	7999	28069	2872	4132	5829	8920	8920
3BL1 h: DS	e	—	—	0.6064	0.8445	—	—	—	—	—
	EX	18	21	100	100	18	21	26	33	33
	IN	3776	5075	27257	52740	2879	4210	5728	8372	8372
SBL	e	2.0062	2.2068	2.5077	3.0091	2.0061	2.2067	2.5076	3.0091	3.0091
	EX	5	5	5	5	5	5	5	5	5
	IN	699	702	686	657	598	611	580	562	562
3BL2	e	—	—	0.6212	0.7575	—	—	—	—	—
	EX	18	21	100	100	18	22	26	34	34
	IN	3767	5079	56559	57383	2880	4481	5758	8971	8971
SGLOB	e	536.0498	318.2929	90.7790	61.7805	533.8325	442.7886	315.4074	75.6387	75.6387
	EX	100	100	100	100	100	100	100	100	100
	IN	57930	58489	58497	58499	95352	97178	96249	97254	97254

O algoritmo sem globalização mostrou grande deficiência na resolução dos pro-

blemas mais difíceis e não resolveu 4 problemas propostos, enquanto o algoritmo **3BL1** com a escolha de Bellavia e Morini para o parâmetro h na estratégia *matrix free* teve melhor desempenho e falhou em somente um problema. Além disso, é interessante notar que entre os testes que divergiram, os resultados dos testes com estratégia de globalização ficaram relativamente muito mais próximos da solução exata do que os resultados obtidos pelo algoritmo sem globalização. Outro dado que devemos ressaltar é o desempenho do algoritmos sem busca linear, que em 6 testes convergiu para uma solução diferente da que esperávamos.

A primeira justificativa para a divergência de alguns testes foi a estagnação do método GMRES com recomeços. Refizemos os testes, alterando o tamanho máximo do ciclo de GMRES para $m = 50$.

Nesse novo teste, cujos resultados também podem ser conferidos na Tabela 2, os Algoritmos **3BL1** e **3BL2** tiveram desempenho satisfatório e resolveram todos os problemas propostos. O algoritmo sem globalização novamente se mostrou ineficiente e divergiu nos mesmos 4 testes do caso anterior. O algoritmo sem busca linear novamente convergiu para outras soluções.

6. Conclusões

Apoiados na idéia proposta por Brown e Saad em [3], desenvolvemos três algoritmos com opções distintas na estratégia de globalização: o primeiro deles com uma estratégia exclusivamente baseada em regiões de confiança, e os outros dois com no máximo três buscas lineares prosseguidas, caso necessário, da estratégia de regiões de confiança. O que difere nesses dois últimos é o critério de aceitação para pontos advindos da estratégia de globalização.

O primeiro algoritmo não obteve um desempenho satisfatório durante a resolução dos testes numéricos, tendo desempenho inferior aos demais, inclusive em relação ao algoritmo sem globalização. Acreditamos que isso se deve principalmente à rigidez dessa estratégia com relação às demais, e, eventualmente, melhores escolhas para o valor do raio inicial podem ser uma saída para contornar esse desempenho insatisfatório. Os demais algoritmos mostraram eficiência e robustez, e nos problemas em que foi necessário o acionamento das estratégias de globalização, demonstraram bom desempenho resolvendo quase todos os testes propostos.

Alguns problemas não foram resolvidos por causa de estagnações causadas pela estratégia de recomeços do GMRES, sendo que alguns problemas só foram resolvidos quando aumentamos o número máximo de iterações em cada ciclo do GMRES. Consideramos que uma saída ideal para este tipo de problema seria o uso de uma estratégia não-constante para o valor de m , esse é um tema para futuras pesquisas. Outra sugestão seria adicionar o uso de pré-condicionadores.

Por fim, gostaríamos de ressaltar o melhor desempenho de nosso algoritmo frente aos algoritmo proposto por Bellavia e Morini em [1]. Acreditamos que esse fato deve-se principalmente à estratégia não-monótona (4.2) adotada para aceitação de pontos.

Agradecimentos

Agradecemos ao *referee* pela leitura e aconselhamento a respeito deste trabalho.

Abstract. In this work, we presented a globally convergent Newton-inexact algorithm for solving nonlinear systems. To ensure global convergence, we use a hybrid strategy combining line search and trust region (dogleg). The linear systems are solving by GMRES(m) method with the matrix-free strategy.

Referências

- [1] S. Bellavia, B. Morini, A globally convergent Newton-GMRES subspace method for system of nonlinear equations, *SIAM J. Sci. Comput.*, **23** (2001), 940-960.
- [2] E.J. Birgin, N. Krejčić, J.M. Martínez, Globally convergent inexact quasi Newton methods for solving nonlinear systems, *Numerical Algorithms* (2003), 249-260.
- [3] P.N. Brown, Y. Saad, Hybrid methods for nonlinear systems of equations, *SIAM Journal of Sci. Stat. Comput.*, **11** (1990), 450-481.
- [4] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.*, **19** (1982), 401-408.
- [5] J.E. Dennis, H.H.W. Mei, Two new unconstrained optimization algorithms which use function and gradient values, *J. Optim. Theory Appl.*, **28** (1979), 453-482.
- [6] J.E. Dennis, R.B. Schnabel, “Numerical Methods for Unconstrained Optimization and Nonlinear Equations”, SIAM, Philadelphia, 1996.
- [7] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Programming Ser.*, **A91** (2002), 201-213.
- [8] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in inexact-Newton methods, *SIAM J. Sci. Comput.*, **17** (1996), 16-32.
- [9] C.T. Kelley, “Iterative Methods for Linear and Nonlinear Equations”, SIAM, Philadelphia, 1995.
- [10] L. Luksán, Inexact trust region method for large sparse system of nonlinear equations, *J. Optim. Theory Appl.*, **81** (1994), 569-590.
- [11] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **7** (1986), 856-869.