

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

PEDRO MACHADO SANTOS ROHDE

## **TRABALHO COMPUTACIONAL**

OTIMIZAÇÃO PELO MÉTODO DE NEWTON

EEL7031 Computação Científica II – prof<sup>º</sup>. Erlon Cristian Finardi

Florianópolis

2017

## Considerações iniciais:

Este trabalho consiste em encontrar o mínimo da função  $f$  de quatro variáveis através do método de Newton para otimização utilizando *MATLAB*. Sendo esse um método iterativo, foram definidos critérios de parada (o número máximo de iterações e a tolerância) e uma solução inicial, a serem explorados a seguir.

O método de Newton consiste simplificadaamente em aproximar uma função quadrática  $f(x)$  para o entorno de um ponto  $x$  a cada iteração e então dar um passo em direção ao ponto mínimo da mesma, que passa a ser o novo ponto a ser aproximado por uma nova função.

Visto que queremos encontrar o ponto mínimo da função, e em pontos de extremos tem-se que as derivadas são zero, devemos buscar um ponto onde a norma do gradiente seja suficientemente próxima de zero.

O método de Newton requer uma aproximação inicial para a solução e também um critério de parada. Neste trabalho foram considerados dois pontos iniciais ( $x^0$ ) distintos,  $(0 \ 0 \ 0 \ 1)^T$  e  $(1 \ 1 \ 1 \ 1)^T$ , com no máximo cem iterações e uma tolerância de  $10^{-8}$  para a norma do vetor gradiente de  $f$  em  $x^k$  ( $\nabla f(x^k)$ ), onde  $k$  é o número da iteração. Para a resolução do sistema linear  $Hf(x^k)p^k = -(\nabla f(x^k))$  foram utilizados três métodos diferentes: eliminação gaussiana, fatoração de Cholesky e gradiente conjugado. Para as medições do tempo de execução de cada método, foi utilizada a média de cinco execuções da parte principal do programa, desconsiderando plotagem de gráficos. A medição se deu com base nas funções *tic* e *toc* do *MATLAB*.

## Eliminação gaussiana:

Na resolução dos sistemas lineares pelo método da eliminação gaussiana, não foi utilizada nenhuma estratégia de pivotamento.

Para o ponto inicial  $(0 \ 0 \ 0 \ 1)^T$ , o método convergiu em 14 iterações, e a solução encontrada foi:  $\mathbf{x}^{14} = \begin{pmatrix} 0.987753175768177 \\ 0.624086561268974 \\ -0.387617363062185 \\ -0.502674942156007 \end{pmatrix}$ .

Neste caso, obteve-se para o valor mínimo da função  $f(\mathbf{x}^{14}) = 26.558945679271037$ , onde a norma euclidiana do gradiente é  $\|\nabla f(\mathbf{x}^{14})\|_2 = 3.117833794943918 \times 10^{-10}$ .

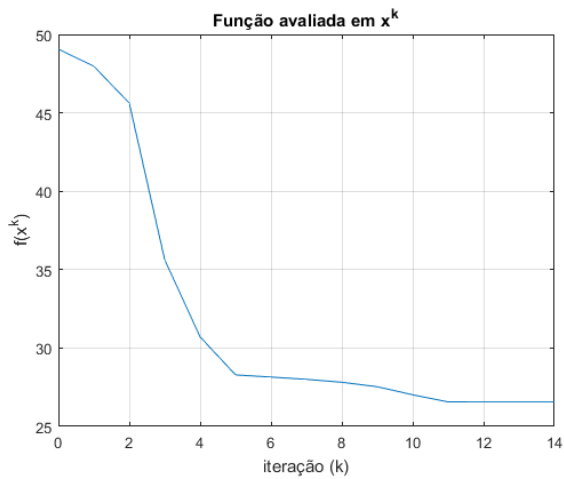
A média de tempo de execução do método (da declaração de variáveis à solução final, sem contar plotagem de gráficos e outros) em cinco execuções foi de 1.2856136 milissegundos.

Para  $\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$ , o método converge em apenas 6 iterações, mesmo o vetor diferença entre a solução inicial e a final ter norma maior que anteriormente. Isso se deve ao fato de que, para esse  $\mathbf{x}^0$ , o tamanho do passo foi 1 em todas as iterações. Uma interpretação geométrica para isso é que a descida no ponto em questão foi mais acentuada do que no anterior.

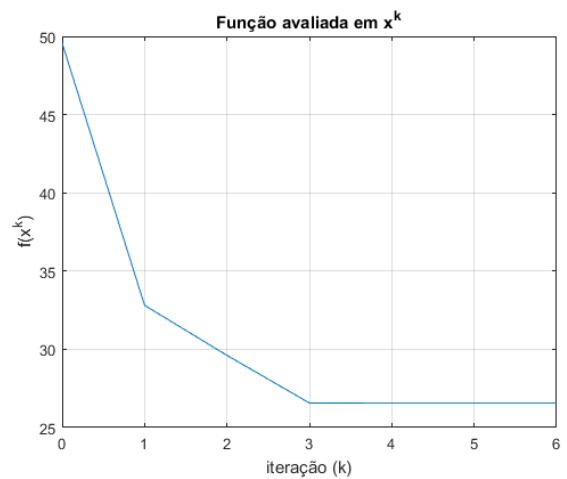
Neste caso,  $f(\mathbf{x}^6) = 26.558945679271037$  é numericamente igual ao caso anterior, apesar da solução final ter sido ligeiramente diferente:  $\mathbf{x}^6 = \begin{pmatrix} 0.987753175769413 \\ 0.624086561269047 \\ -0.387617363039680 \\ -0.502674942201398 \end{pmatrix}$ . O gradiente final foi  $\|\nabla f(\mathbf{x}^6)\|_2 = 1.480624191212825 \times 10^{-9}$ .

A média de tempo de execução foi de 0.949563 milissegundos.

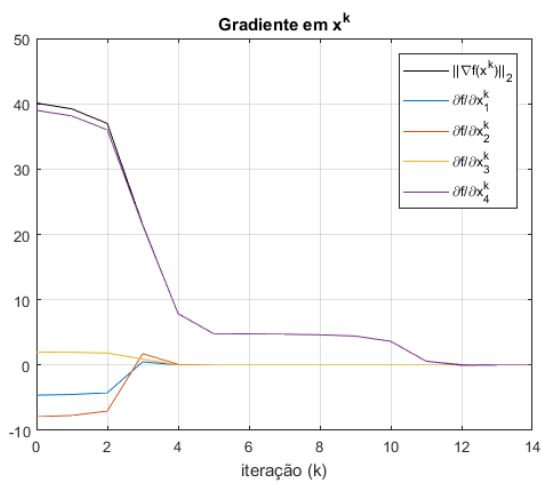
A norma euclidiana da diferença entre as duas soluções finais foi  $5.067868325562979 \times 10^{-11}$ , ou seja, elas ficaram bastante próximas.



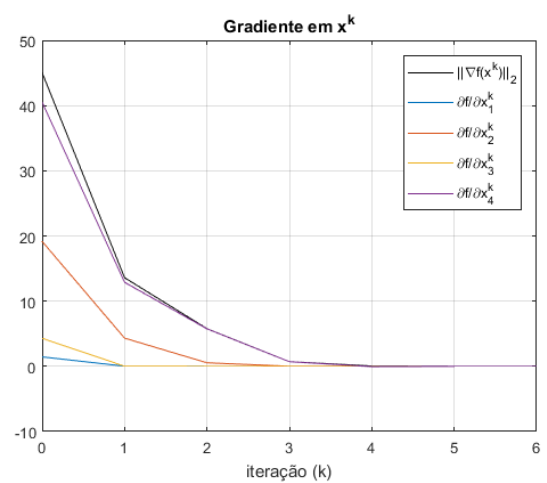
$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



$$x^0 = (1 \ 1 \ 1 \ 1)^T$$

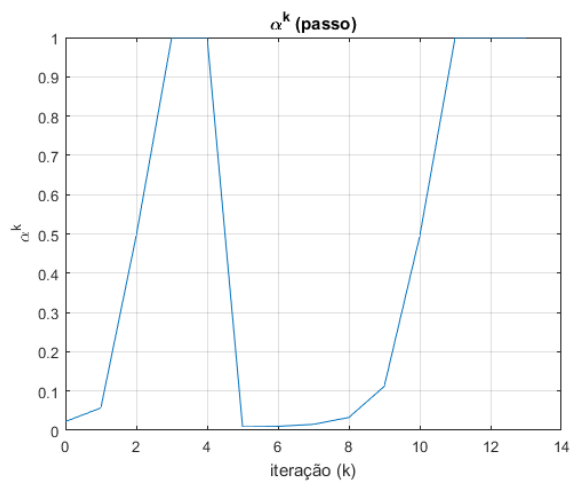


$$x^0 = (0 \ 0 \ 0 \ 1)^T$$

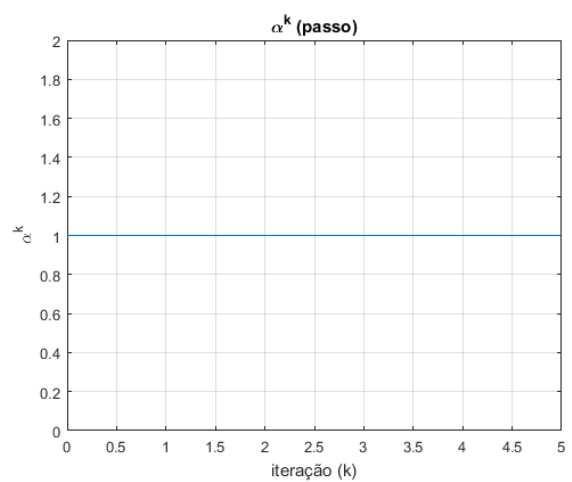


$$x^0 = (1 \ 1 \ 1 \ 1)^T$$

Observa-se que, com os valores altos para as componentes do gradiente para  $x^0 = (1 \ 1 \ 1 \ 1)^T$ , tem-se uma “descida mais acentuada”, o que favorece a convergência.



$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



$$x^0 = (1 \ 1 \ 1 \ 1)^T$$

O tamanho do passo foi 1 (máximo) em todas as iterações para  $x^0 = (1 \ 1 \ 1 \ 1)^T$ .

## Fatoração de Cholesky:

Para  $\mathbf{x}^0 = (0 \ 0 \ 0 \ 1)^T$ , o método convergiu em 14 iterações, assim como usando a eliminação gaussiana. A solução encontrada e a norma do gradiente final foram também as mesmas,

$$\mathbf{x}^{14} = \begin{pmatrix} 0.987753175768177 \\ 0.624086561268974 \\ -0.387617363062185 \\ -0.502674942156007 \end{pmatrix}, \quad f(\mathbf{x}^{14}) = 26.558945679271037 \quad \text{e}$$
$$\|\nabla f(\mathbf{x}^{14})\|_2 = 3.117833794943918 \times 10^{-10}.$$

A diferença entre os dois métodos de resolução do sistema linear se deu no tempo de execução: para esse ponto inicial, a média das execuções foi de 2.1599092 milissegundos. Essa piora no desempenho com relação ao tempo se deu pois.

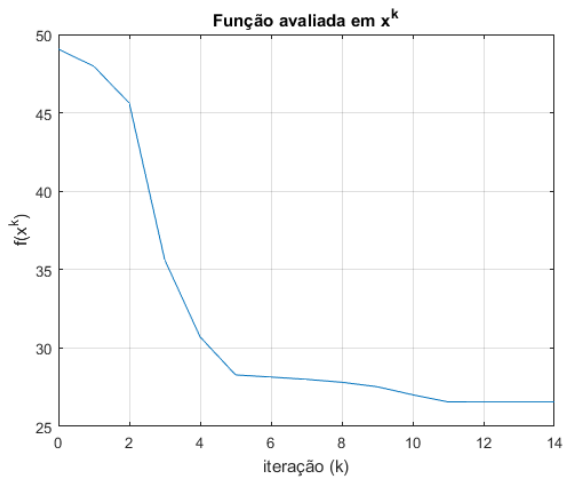
Com  $\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$ , assim como para o ponto anterior, obteve-se um desempenho igual ao da eliminação gaussiana para todos os parâmetros menos o tempo. Após seis iterações, os resultados finais foram:

$$\mathbf{x}^6 = \begin{pmatrix} 0.987753175769413 \\ 0.624086561269047 \\ -0.387617363039680 \\ -0.502674942201398 \end{pmatrix}, \quad f(\mathbf{x}^6) = 26.558945679271037 \quad \text{e} \quad \|\nabla f(\mathbf{x}^6)\|_2 = 1.480624191212825 \times 10^{-9}.$$

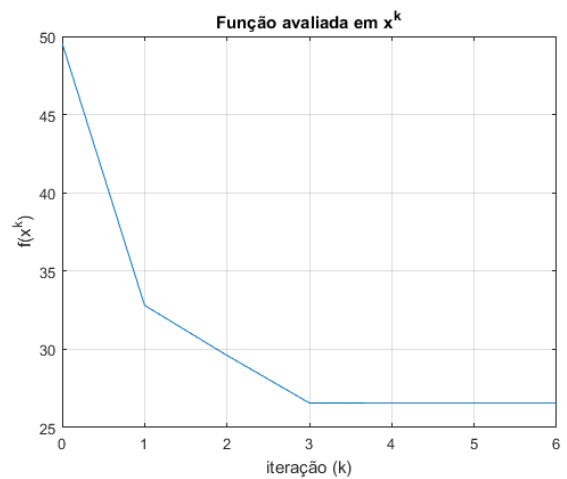
Novamente, a média de tempo de execução foi maior que quando se usou eliminação gaussiana, 1.1120712 milissegundos.

Por serem ambos métodos diretos para resolução de sistemas lineares, a eliminação gaussiana e a fatoração de Cholesky produzem sempre resultados precisos (com uma pequena margem de erro devido a arredondamentos, desprezível aqui), por isso acabam sendo numericamente iguais. A diferença no desempenho temporal entre os dois se deve ao fato de a fatoração Cholesky ter um passo a mais que é a fatoração da matriz na forma  $LL^T$ . O método seria mais indicado se a matriz  $A$  (de  $Ax = b$ ; no caso deste trabalho,  $A$  é a hessiana), fosse sempre a mesma, não sendo necessária uma fatoração a cada iteração. Mas, com  $\mathbf{x}$  variando a cada iteração,  $\mathbf{H}f(\mathbf{x})$  também varia.

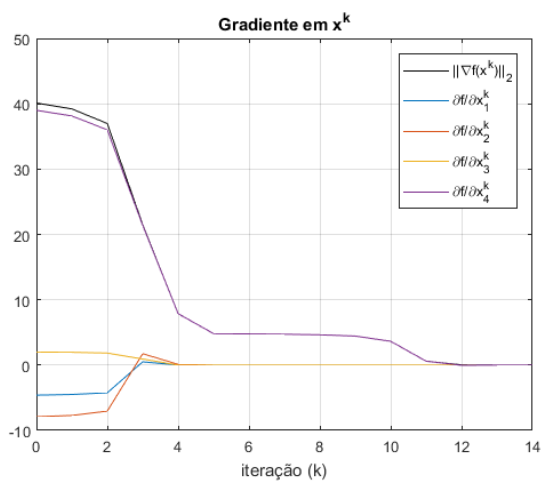
Os gráficos da fatoração de Cholesky são semelhantes ao da eliminação gaussiana pelos motivos já explicados acima.



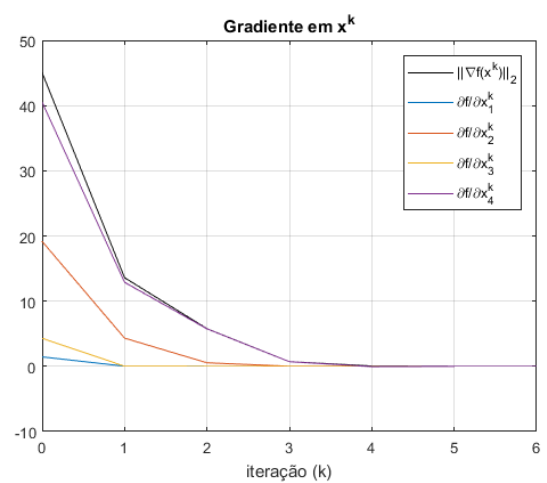
$$\mathbf{x}^0 = (0 \ 0 \ 0 \ 1)^T$$



$$\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$$

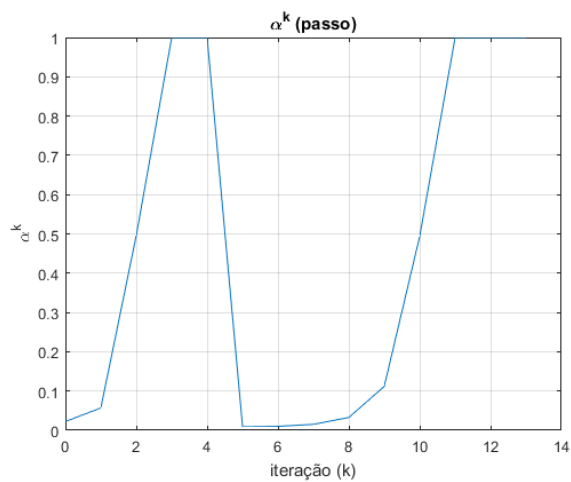


$$\mathbf{x}^0 = (0 \ 0 \ 0 \ 1)^T$$

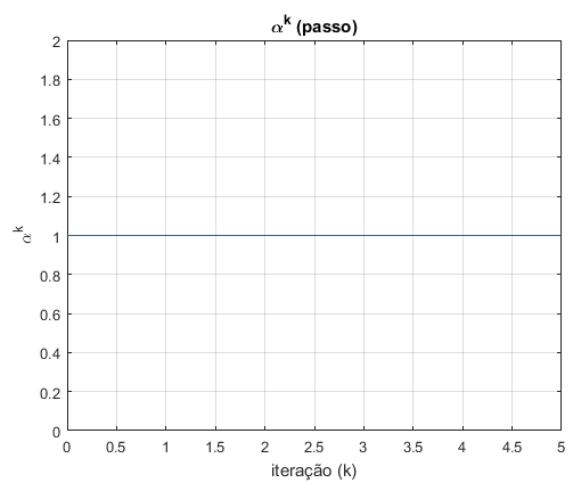


$$\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$$

Observa-se que, com os valores altos para as componentes do gradiente para  $\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$ , tem-se uma “descida mais acentuada”, o que favorece a convergência.



$$\mathbf{x}^0 = (0 \ 0 \ 0 \ 1)^T$$



$$\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$$

O tamanho do passo foi 1 (máximo) em todas as iterações para  $\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$ .

## Gradiente conjugado:

O gradiente conjugado é um método iterativo de resolução de sistemas lineares, portanto, não necessariamente produz uma solução exata. Foi considerada uma tolerância de  $10^{-5}$  para a norma do resíduo e uma solução inicial  $\mathbf{x}^0 = (0 \ 0 \ 0 \ 0)^T$  na resolução do sistema a cada iteração.

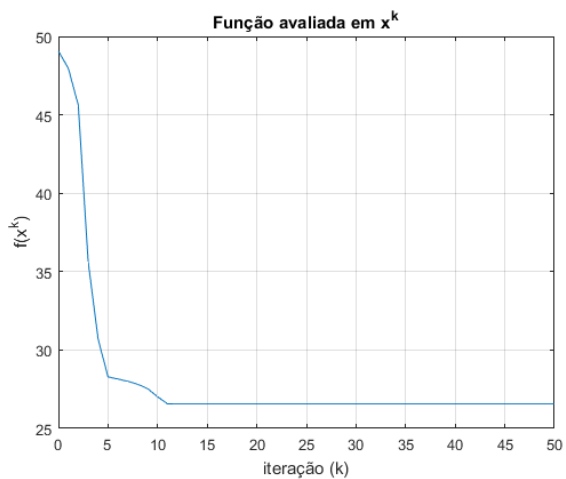
Para a solução inicial  $\mathbf{x}^0 = (0 \ 0 \ 0 \ 1)^T$ , o método de Newton convergiu em 50 iterações para  $\mathbf{x}^{50} = \begin{pmatrix} 0.987753175752386 \\ 0.624086561270288 \\ -0.387617362518536 \\ -0.502674942166203 \end{pmatrix}$ , com  $f(\mathbf{x}^{50}) = 26.558945679271030$  e  $\|\nabla f(\mathbf{x}^{50})\|_2 = 1.076528294884233 \times 10^{-9}$ .

A média do tempo de execução foi de 5.151691 milissegundos.

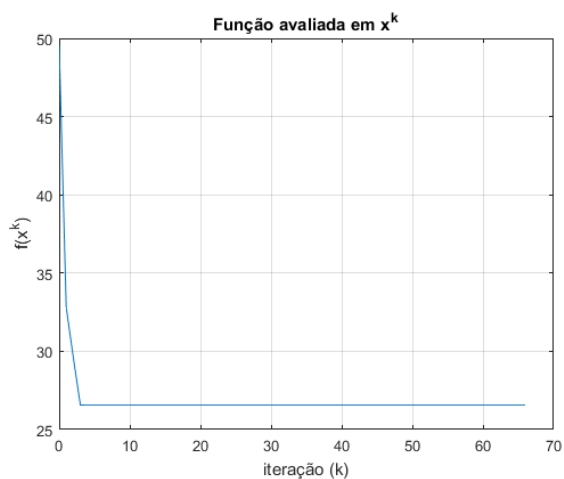
Com  $\mathbf{x}^0 = (1 \ 1 \ 1 \ 1)^T$ , a convergência se deu em 66 iterações até encontrar  $\mathbf{x}^{66} = \begin{pmatrix} 0.987753175670859 \\ 0.624086561134700 \\ -0.387617359877700 \\ -0.502674942277963 \end{pmatrix}$ ,  $f(\mathbf{x}^{66}) = 26.558945679271030$  e  $\|\nabla f(\mathbf{x}^{66})\|_2 = 7.407679674778141 \times 10^{-9}$ . O tempo de execução médio foi de 5.9492656 milissegundos.

Apesar de demorar mais tempo e precisar de mais iterações, quando se utilizou o método do gradiente conjugado com tolerância  $10^{-5}$ , obteve-se soluções cujo  $f(\mathbf{x})$  foi menor do que com os métodos vistos anteriormente (a diferença foi da ordem de  $10^{-15}$ ).

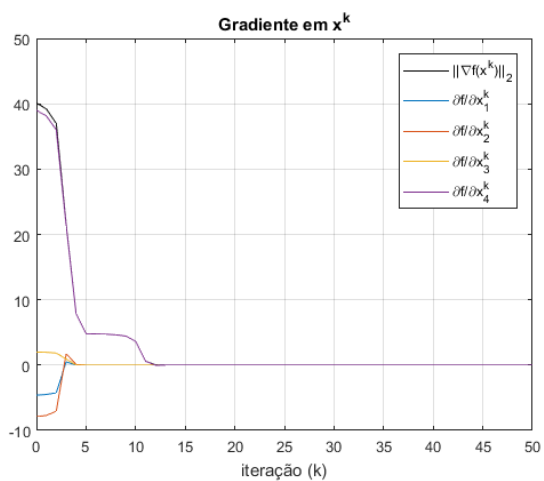
Diferentes tolerâncias para o resíduo no gradiente conjugado produzem diferentes resultados. Para  $10^{-8}$ , a solução final encontrada é numericamente igual às encontradas pelos outros métodos. Para tolerâncias menores ainda, não se observa melhora significativa no desempenho. Para tolerâncias maiores, o método de Newton converge em um número cada vez maior de iterações.



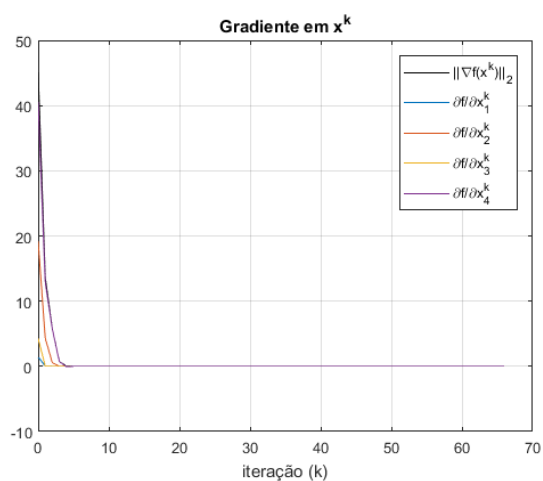
$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



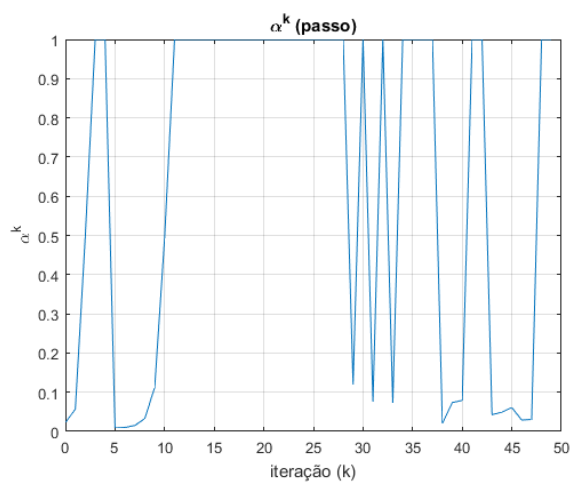
$$x^0 = (1 \ 1 \ 1 \ 1)^T$$



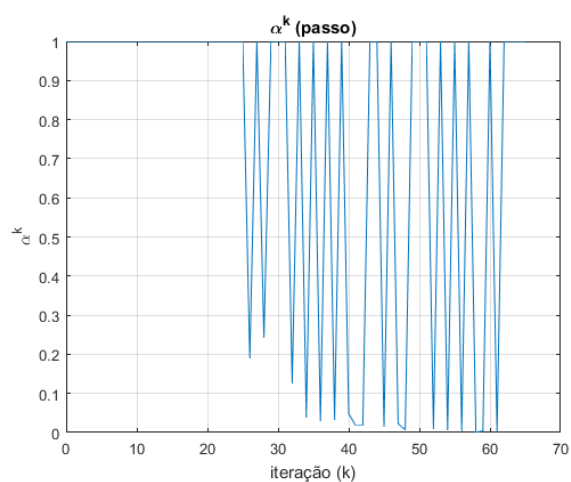
$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



$$x^0 = (1 \ 1 \ 1 \ 1)^T$$



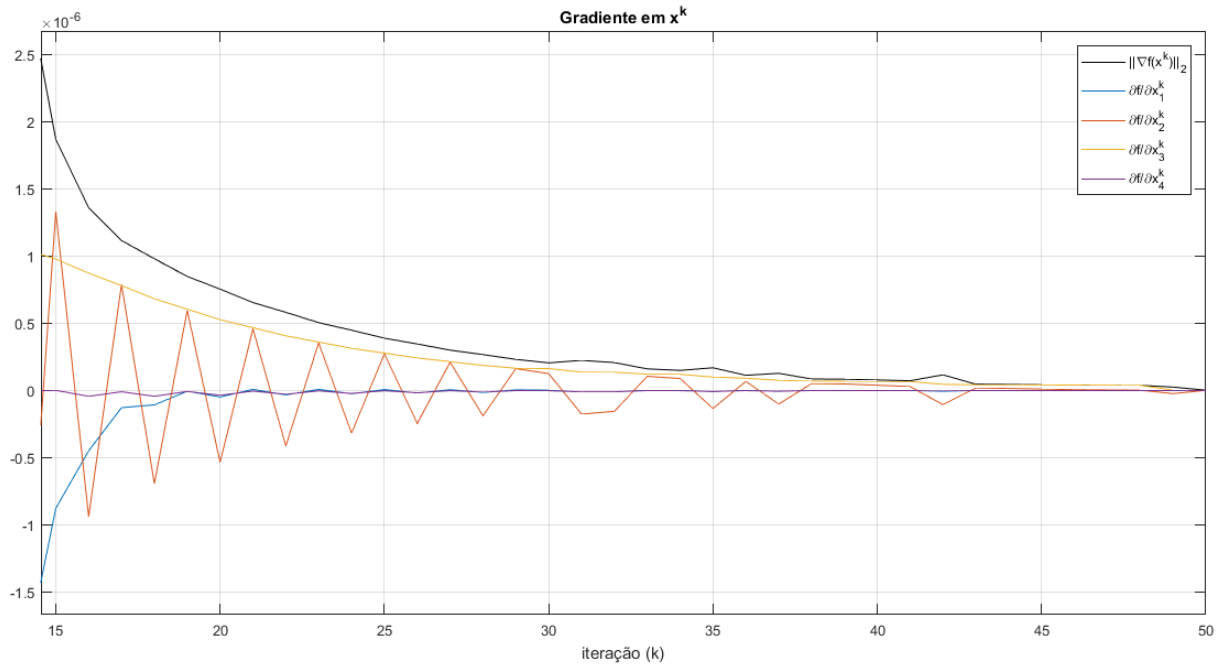
$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



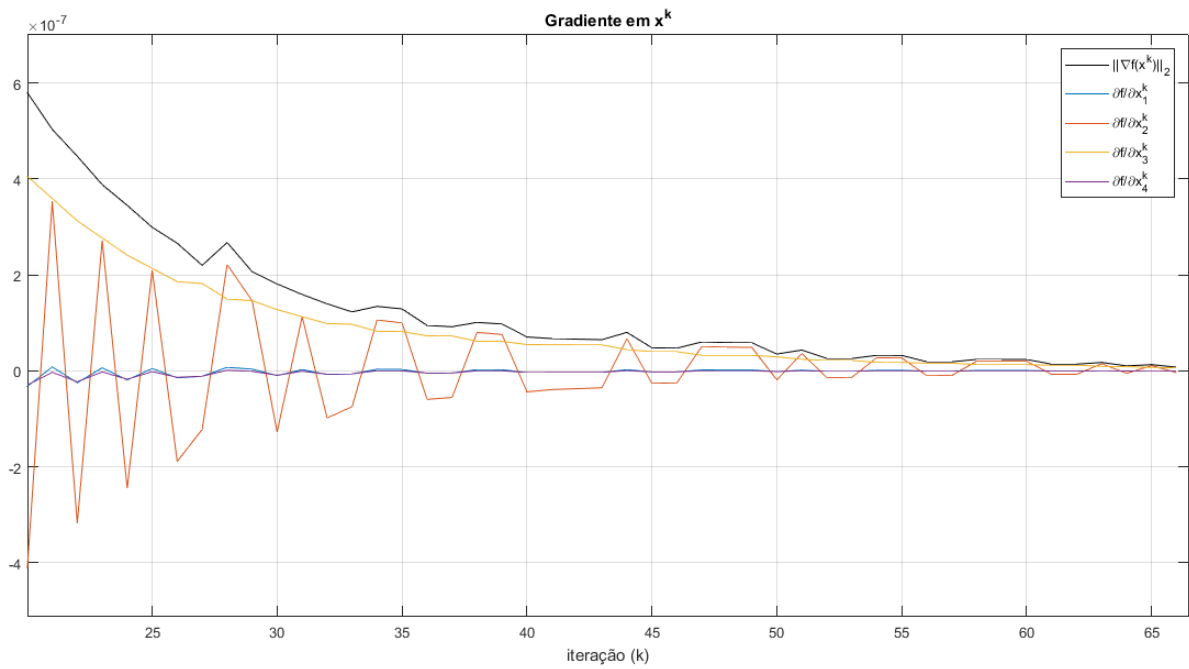
$$x^0 = (1 \ 1 \ 1 \ 1)^T$$



À primeira vista, olhando para os gráficos dos gradientes, parece que  $\|\nabla f(x)\|_2$  vai a zero muito antes do algoritmo parar. Essa percepção se dá pelo fato de a norma do gradiente ter uma convergência muito lenta a partir de certa iteração. Apesar de o método não alcançar o critério de parada,  $f(x)$  permanece praticamente inalterado nessas iterações. Esse efeito pode ser melhor observado nos gráficos a seguir:



$$x^0 = (0 \ 0 \ 0 \ 1)^T$$



$$x^0 = (1 \ 1 \ 1 \ 1)^T$$

## Considerações finais:

Os métodos diretos (eliminação gaussiana e Cholesky) produzem soluções exatas, portanto, a cada iteração tem-se o mesmo sistema linear a ser resolvido por ambos, resultando sempre na mesma solução, o que se reflete no mesmo resultado final, só variando o tempo de execução. O resultado da resolução dos sistemas pelo método do gradiente conjugado depende da tolerância estabelecida previamente e, portanto, pode variar.

Para o sistema linear considerado neste trabalho (quatro equações e quatro variáveis), os métodos diretos acabaram se provando mais eficientes em termos de tempo (mais rápidos) e esforço computacional (menos iterações) do que o iterativo (gradiente conjugado), apesar de isso não ser necessariamente verdadeiro sempre. Para sistemas maiores e com grande percentual de elementos nulos, os métodos iterativos são mais indicados, pois o erro devido ao arredondamento é reduzido. Para os diretos, quanto mais operações são realizadas, maior vai ficando o erro de arredondamento acumulado, visto que o computador só é capaz de realizar operações com um número finito de casas decimais.

Apesar de as medições de tempo terem sido todas feitas na mesma máquina, certos processos incontrolláveis do computador podem ter influenciado a execução do programa. A média aritmética de cinco medições foi usada para tentar-se contornar essa possibilidade. Os resultados podem variar de acordo com o processador utilizado, o sistema operacional, a versão do *MATLAB* (ou até mesmo a linguagem escolhida para a programação), a memória *RAM* disponível e inúmeros outros fatores.

Por fim, para este problema específico, a ordem de desempenho dos métodos com relação ao tempo e ao número de iterações foi: eliminação gaussiana, fatoração de Cholesky e gradiente conjugado, apesar do último ter achado o melhor ponto de mínimo.