MAP3121 - Exercício programa 1 - 2018
GPS e o Método de Newton

O Sistema de Posicionamento Global

O Sistema de Posicionamento Global (GPS, sigla do nome em inglês) é um sistema de navegação formado por uma rede de 24 satélites orbitando a Terra e transmitindo sinais para os receptores GPS, que usam triangulação para calcular a localização do usuário. Na descrição a seguir, o sistema de coordenadas tem sua origem O no centro da Terra, o eixo Oz positivo aponta na direção do Pólo Norte, o plano Oxy é o plano do equador com o eixo Ox positivo cortando o meridiano de Greenwich e o eixo Oy positivo corta o meridiano de longitude 90°E.

Um dado satélite i transmite a sua posição atual (x_i, y_i, z_i) e o instante te_i de emissão do sinal. O receptor GPS grava esta informação juntamente com o instante tr_i da recepção do sinal. Entretanto, o relógio do receptor é menos preciso do que o relógio do satélite, havendo um erro de sincronia T. O instante correto da recepção é $tr_i - T$. Se $T > 0$ o relógio do receptor está adiantado e se $T < 0$ ele está atrasado.

Portanto, o tempo que o sinal leva do satélite ao receptor é $t_i - T$, onde $t_i = tr_i - te_i$ é o lapso de tempo aparente, e o receptor deve estar na superfície da esfera de raio $c(t_i - T)$ e centro (x_i, y_i, z_i), onde $c = 299792458\ m/s$ é a velocidade da luz no vácuo. As coordenadas (x, y, z) do receptor devem então satisfazer a relação

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = (w_i - w)^2$$

onde $w_i = ct_i$ e $w = cT$ (ao multiplicarmos por c, ficamos apenas com dimensões espaciais).

Em qualquer instante, dados de 5 a 8 satélites são obtidos pelo receptor em qualquer ponto da terra. Então, assumindo que temos dados de $n > 4$ satélites, as 4 incógnitas (x, y, z, w) devem satisfazer as equações

$$r_i(x, y, z, w) = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - (w_i - w)^2 = 0, \qquad (1)$$

$1 \leq i \leq n$, que formam um sistema não-linear sobredeterminado. Como as medidas estão sujeitas a erros, este sistema em geral não tem solução (se não houvesse erros, bastaria resolver 4 das n equações).

Mínimos quadrados não-lineares

Uma maneira de se resolver o problema acima é tratá-lo como um problema de mínimos quadrados, procurando-se minimizar a soma dos quadrados dos resíduos r_i, ou seja, determinando-se $\bar{x}$, $\bar{y}$, $\bar{z}$ e $\bar{w}$ que minimizam

$$g(x,y,z,w) = \sum_{i=1}^{n} r_i(x,y,z,w)^2.$$

Uma condição necessária que um ponto de mínimo deve satisfazer é

$$\nabla g(x,y,z,w) = 0 \tag{2}$$

onde ∇g é o gradiente de g em relação a x, y, z e w. A condição acima é um sistema não-linear de 4 equações e 4 incógnitas, que será resolvido usando o método de Newton.

Mínimos quadrados lineares

Suponha que as equações (1) são exatas. Se subtrairmos a equação n da equação i, os termos quadráticos cancelam e portanto das relações

$$r_i(x,y,z,w) - r_n(x,y,z,w) = 0, \quad 1 \leq i \leq n-1 \tag{3}$$

obtemos um sistema linear sobredeterminado para as incógnitas que pode ser resolvido no sentido de mínimos quadrados como foi (será ou deveria ser) visto em aula.

Se A denota a matriz do sistema linear, $\mathbf{x}$ o vetor incógnita e $\mathbf{b}$ o lado direito do sistema linear, a solução de $A\mathbf{x} = \mathbf{b}$ no sentido de mínimos quadrados é a solução do sistema normal $A^TA\mathbf{x} = A^T\mathbf{b}$. Esta solução pode ser usada como uma aproximação em si para a localização usando GPS, e também como aproximação inicial para a resolução de (2) usando o método de Newton.

Tarefa

Escreva um programa para resolver um sistema não-linear de m equações e m incógnitas usando o método de Newton. Teste o programa com os exemplos apresentados na descrição do método de Newton. Use o programa para calcular a posição e o erro de sincronização de um receptor GPS a partir de dados de n satélites, resolvendo-se o sistema não linear (2). A aproximação inicial deve ser obtida da solução do sistema normal derivado de (3) usando-se a decomposição LU de A^TA.

Os dados de entrada devem ser lidos de um arquivo cujo formato é: na primeira linha, há um inteiro especificando o número de satélites. Nas linhas abaixo temos uma matriz $n \times 4$ contendo os valores x_i, y_i, z_i e w_i em metros. O programa deve imprimir os dados de entrada bem como a aproximação inicial e o número de iterações usado pelo Método de Newton, os valores finais para as coordenadas do receptor e o erro de sincronização (não se esqueça de dividir $\bar{w}$ por c antes de imprimir).

Teste o seu programa com os dados do arquivo disponibilizado *dadosgps.txt*. A partir dos resultados determine a latitude, a longitude e a elevação (coordenadas geográficas) do receptor e veja onde ele está no globo terrestre. Note que os dados são apresentados com muitos algarismos. Isto é necessário devido à alta sensibilidade do GPS a erros. Para ler mais sobre o Sistema de Posicionamento Global, consulte as referências

S. Alves, *A matemática do GPS*, Revista do Professor de Matemática **59** (2006), pp 17–26 (disponível em www.rpm.org/conheca/gps.pdf)

G. Nord, D. Jabon, and J. Nord, *The global positioning system and the implicit function theorem*, SIAM Review **40**(3) (1998), pp 692–696

Método de Newton para funções de várias variáveis

Vamos descrever o método de Newton para determinação de raízes de funções $F(x)$ de R^n em R^n. Como no caso unidimensional, parte-se de uma aproximação inicial $x^{(0)}$ para o valor $\bar{x} \in R^n$ tal que $F(\bar{x}) = 0$ e calcula-se a sequência

$$x^{(k+1)} = x^{(k)} - J_F^{-1}(x^{(k)})F(x^{(k)})$$

onde $J_F(x)$ é a matriz Jacobiana de F avaliada no ponto x. Para F de R^n em R^n,$J_F(x)$ é dada por:

$$J_F(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} ,$$

onde $f_1(x), f_2(x), ..., f_n(x)$ são as componentes de $F(x)$.

Na descrição da iteração do método de Newton aparece a inversa de $J_F(x)$. Pode-se evitar a necessidade de inverter $J_F(x)$ reescrevendo a iteração na forma:

$$J_F(x^{(k)})\ (x^{(k+1)} - x^{(k)}) = -F(x^{(k)}) \quad .$$

Assim, a cada passo do método de Newton resolve-se o sistema linear

$$J_F(x^{(k)})\ c^{(k)} = -F(x^{(k)})$$

e calcula-se a nova aproximação como $x^{(k+1)} = x^{(k)} + c^{(k)}$. Quando F é de classe C^2 e a matriz Jacobiana é não singular na raiz de F (ou seja, tem determinante não nulo), pode-se mostrar que o método de Newton converge quadraticamente para a raiz de F, desde que $x^{(0)}$ seja escolhido suficientemente próximo da raiz.

Você encontra mais informações sobre o método de Newton n-dimensional nos livros de Burden e Faires (veja as informações gerais do curso) ou Isaacson e Keller (Analysis of Numerical Methods), disponíveis na Biblioteca do IME, ou mesmo através de uma busca na Internet.

Programando o método de Newton

A implementação do método de Newton requer rotinas para a avaliação de $F(x)$ e da matriz jacobiana $J_F(x)$. Além disso, deve-se resolver um sistema linear a cada passo. Para a solução do sistema linear usaremos a decomposição LU da matriz. Descrevemos abaixo o algoritmo (com pivotação parcial) para o cálculo da decomposição LU de uma matriz A (eventualmente permutada), onde L é triangular unitária (ou seja, tem 1's na diagonal) inferior e U é triangular superior.

Dados n e uma matriz A $(n \times n)$ temos:

- Para k de 1 a n faça
  - para i de k a n faça
    $a(i,k) = a(i,k) - \sum_{j=1}^{k-1} a(i,j) * a(j,k)$
  - Determine $l \geq k$ tal que $|a(l,k)| = \max_{k \leq i \leq n} |a(i,k)|$
  - defina $p(k) = l$
  - se $k \neq p(k)$ troque linhas k e $p(k)$ da matriz A
  - para j de k+1 a n faça
    $a(k,j) = a(k,j) - \sum_{i=1}^{k-1} a(k,i) * a(i,j)$
    $a(j,k) = a(j,k)/a(k,k)$

Observações:

- Ao final do algoritmo a matriz L tem seus valores abaixo da diagonal principal armazenados nas posições correspondentes de A (lembre-se que a diagonal de L é composta de $1's$).
- A matriz U tem seus valores da diagonal principal e acima desta armazenados nas posições correspondentes de A
- A decomposição LU calculada corresponde à matriz A permutada. As permutações realizadas estão armazenadas no vetor p definido no algoritmo.
- Lembre-se que ao final do algoritmo a matriz A foi modificada. Caso esta ainda seja necessária, uma cópia sua deve ser anteriormente salva.
- Somatórios de 1 a 0 e loops de $n+1$ a n devem ser entendidos como vazios

Voce deve implementar também um procedimento para resolver um sistema $Ax = b$, que utilize a decomposição LU calculada. Lembre-se que o vetor b deve ser permutado correspondentemente (para tal use o vetor de permutações p).

Testes Iniciais

- Use seu código para determinar o ponto de mínimo da função $F(x,y) = (x-2)^2 + (y-3)^2$, calculando para tanto o ponto onde seu gradiente se anula. (Quantas iterações do método de Newton são necessárias para convergência?)

- Dada a função $F(x_1, x_2, x_3, x_4) = (4x_1 - x_2 + x_3 - x_1x_4, -x_1 + 3x_2 - 2x_3 - x_2x_4, x_1 - 2x_2 + 3x_3 - x_3x_4, x_1^2 + x_2^2 + x_3^2 - 1)$, determine a raiz que se obtém pelo método de Newton tomando $x = (1,1,1,1)$ como valor inicial.

- Utilize o método de Newton para determinar solução do sistema $n-1 \times n-1$, cujas equações são

$$-x_{i-1} + 2x_i - x_{i+1} = \frac{e^{x_i}}{n^2} \ , \quad i = 1, ..., n-1 \quad ,$$

  com $x_0 = x_n = 0$, a partir da aproximação inicial nula. (Teste para $n = 20, 40$ e 80)

Instruções

As analises e resultados obtidos devem ser organizados em um relatório que deve minimamente discutir os problemas estudados e os resultados obtidos.

- O exercício deve ser feito em Python 3.x (o mesmo usado nos cursos de MAC, veja mais detalhes em https://panda.ime.usp.br/panda/python).

- O exercício pode ser feito em duplas, sempre com alguém da mesma área, não necessariamente da mesma turma.

- Apenas um aluno deve entregar o exercício, destacando no relatório e código o nome de ambos os alunos.

- A entrega deve conter o relatório (em .pdf), contendo a análise do problema estudado, e o código usado para as simulações computacionais (arquivos .py). A entrega pode ser feita em um arquivo compactado único.

O seu código deve estar bem comentado e estruturado. A entrada e saída devem ser feitas de forma a ajudar o usuário a executar o programa e deve facilitar a análise dos resultados. Inclua qualquer arquivo adicional necessário para o seu programa no arquivo compactado a ser entregue. Como o seu programa terá que ler arquivos de entrada, considere que os mesmos encontram-se na mesma pasta executável, ou solicite o caminho/nome do arquivo ao usuário.

Você deve resolver tanto os exercícios relativos à aplicação, descritos na parte de GPS, quanto os testes descritos na seção anterior de Testes Iniciais.