Relatório do ep3 - MAC300

Decomposição QR para o cálculo de mínimos quadrados

Nomes:	**Nusp:**
Fernanda de Camargo Magano	8536044
Eduardo Delgado Coloma Bier	8536148

São Paulo
2014

Problema de mínimos quadrados, decomposição QR usando reflexões

- O uso de refletores é mais eficiente do que o uso de rotações. As matrizes estão sendo consideradas como retangulares, o que é mais abrangente, pois o algoritmo serve também para matrizes quadradas.
- Uma das coisas implementadas na busca da eficiência foi: a matriz R é guardada na própria A, enquanto a Q não será guardada. Não é necessário guardar a Q, apenas os vetores u de cada iteração e o gama. Com isso já se consegue calcular o valores de Q, sem guardá-la explicitamente. O u(1) é armazenado na primeira coluna de A, u(2) na segunda e assim sucessivamente. O tau também é armazenado na própria matriz A.
- Para evitar que ocorram overflows ou underflows foi implementado o seguinte: procurar o maior elemento da matriz e dividir todos os elementos de A e de b por esse valor máximo. Portanto, foi feito um reescalamento.
- O algoritmo é usado tanto para posto completo, quanto para incompleto. Para isso, é usado o pivoteamento de colunas. Isso garante a troca das mesmas, colocando na frente a que tiver maior norma. Assim, num determinado momento do algorimo, se a matriz for de posto incompleto, a máxima norma das colunas será menor do que um certo epsilon. Esse é momento em que se descobre o posto da matriz e o algoritmo de decomposição QR termina.
- Ainda pensando em eficiência, o código foi implementado com orientação a linhas, mantendo a essência do algoritmo de ir zerando as colunas, colocando a norma de cada coluna no primeiro elemento da mesma. Mesmo o cálculo de normas foi feito orientado a colunas. Cada norma foi sendo calculada parcialmente, percorrendo cada linha para tal finalidade.
- Outro ponto importante para o desempenho do algoritmo foi que as normas calculadas para as colunas são aproveitadas a cada passo, isto é, tem o custo inicial fixo de calcular todas as normas. Contudo, a cada iteração, essas normas previamente calculadas são reaproveitadas. O custo com essa melhoria é de 2(m-1), enquanto antes era 2(m-1)(n-1) se fosse calcular a cada vez.

----------Formato do arquivo----------

Os arquivos para leitura devem ter o seguinte formato:
Na primeira linha estará o inteiro n (número de linhas) e m (número de colunas);
Depois haverá n X m linhas com três números (os dois primeiros são inteiros: a linha e a coluna da matriz e o terceiro é um double – o valor do elemento)*;*
Por último, n linhas contendo um inteiro e um double, indicando a posição e o valor dos elementos do vetor b.

----------Testes usando QR e polinômios:----------

Para montar o polinômio de grau 5 pode-se ter em mente que se A é uma matriz e p(x) é o polinômio, “substitui-se” os x's por A, chegando numa matriz modificada que será usada no programa para cálculo dos mínimos quadrados. Assim, testa-se com outros polinômios, fazendo o mesmo processo de cálculo (substituindo x por A nos polinômios usados para a aproximação). O cálculo do resíduo vai mostrar se o polinômio usado fez uma aproximação boa ou razoável da almejada.

Se x é um vetor contendo os coeficientes de um polinômio p, os mesmos são normalizados para tornar o polinômio mônico. Monta-se a "companion matrix" A (sua primeira linha contém os opostos dos coeficientes de -an-1 … a0 e a subdiagonal possui 1's – o restante é zero). Assim, os autovalores da matriz A são calculados por QR. Esses são exatamente as raízes do polinômio.

----------Cálculo do polinômio:----------

- Imagine um polinômio de grau cinco. Então escolhemos 15 pontos do mesmo. Sejam eles: (x1, y1) ...(xm, ym), sendo m=15. Deseja-se conseguir aproximações usando mínimos quadrados
- O objetivo é encontrar polinômios de grau p que melhor se ajuste a esses dados. Eles são da forma yi = cp*xi^p + c(p-1) * xi^(p-1) ...+ c1x1 + c0x0 + resíduo

```
b = | y1|        A = |x1^p …....   x1 1   |
    | .  |            |x2^p.....   x2  1   |
    | .  |            |:                   |
    | .  |            |.                   |
    | yn |            | xm^p.....   xm 1|
```

- Monta-se a matriz A e b e resolve por QR
- A matriz A, como indicada, é construída usando potências de xi. É uma forma de conseguir uma base que melhor aproxima. Mais adequada do que a canônica, por exemplo.
- Os x's obtidos pelo QR serão usados na montagem do polinômio desejado
- Depois calcula-se o resíduo e é verificado se a aproximação foi boa ou não.
- Seja um polinômio: $P(x) = x^5 + 2x^4 + 1x^3 + 3x^2 - 7x + 6$

- Considere os seguintes quinze pontos abaixo:

$P(x) = x^5 + 2x^4 + 1x^3 + 3x^2 – 7x + 6$

Pontos:

(0, 6)
(1, 6)
(-1, 16)
(1,5 ; 23,34375)
(2 ; 76)
(2,3 ; 138,26863)
(2,8 ; 326,90688)
(3,3 ; 680,04513)
(4 ; 1626)
(5 ; 4546)
(5.5 ; 7087,59375)
(6; 10656)
(7 ; 22056)
(8; 41614)
(10 ; 121,236)

Então o b é:

b:

6
6
16
23.34375
76
138.26863
326.90688
680.04513
1626
4546
7087.59375
10656
22056
41614
121236

O A será montado de acordo com o grau do polinômio que se deseja aproximar.
Por exemplo, se quisermos usar um polinômio de grau cinco para fazer a aproximação, o A será igual a:

0	0	0	0	0	1
1	1	1	1	1	1
-1	1	-1	1	-1	1
7.59375	5.0625	3.375	2.25	1.5	1
32	16	8	4	2	1
64.36343	27.9841	12.167	5.29	2.3	1
172.10368	61.4656	21.952	7.84	2.8	1
391.35393	118.5921	35.937	10.89	3.3	1
1024	256	64	16	4	1
3125	625	125	25	5	1
5032.84375	915.0625	166.375	30.25	5.5	1
7776	1296	216	36	6	1
16807	2401	343	49	7	1
32768	4096	512	64	8	1
100000	10000	1000	100	10	1

O resíduo obtido foi:
Resíduo = 2652.378247

----------Perturbações em matrizes ----------

- Foi testada uma matriz dois por dois, para ser melhor de visualizado o que estava acontecendo;
- A ideia foi perturbar o vetor b em apenas 0.01, mas como A é mal-condicionada, perturbar na direção do maxmag pode causar estragos na solução;
- A matriz A escolhida foi a seguinte:
  25 28
  10 12
- Essa matriz foi construída de modo que as colunas são quase LD. Portanto, é mal-condicionada. A pequena perturbação em b é feita na direção do maxmag de A que é [1 1]^t
- A norma infinita de ||delta b|| / ||b|| = 0.01/53 = 1.887 x 10^{-4}
- A norma infinita de ||delta x|| / ||x|| = 1.28/2.767241 = 0.4625
- Fica visível de que a alteração em b é pequena, mas o erro relativo em x tem proporção maior (enquanto no b é da ordem de 10^{-4} a alteração em x é de 10^{0});

- Os arquivos em que esses testes são realizados são: **ill_conditioned** e **ill_condiotioned_perturbado**
- A solução de x por mínimos quadrados do primeiro arquivo foi:
  -1.000000
  2.767241
  A do segundo:
  -2.280000
  3.886810
  Portanto, o delta x é [-1.28 1.119569] transposto
- Esse exemplo mostra a interferência do número de condição num sistema e como mudanças escolhidas cuidadosamente podem causar alterações em x, principalmente se for na direção de máxima magnificação da matriz.

----------Observações----------

- O programa foi feito em C;
- Alguns testes estão nos arquivos: identidade, ill_conditioned, ill_conditioned_perturbado, triang, estima_pol_5, a_2x1, a_3x2, a_2x2, nula, testa_eps;
- Para compilar e testar os arquivos basta digitar:
  gcc -lm EP3.c -o EP3 <nome de um dos arquivos acima> -v
  Lembrando-se de que a opção -v é opcional (deve ser colocada apenas se quiser imprima várias informações, como a matriz R, a matriz reescalada, normas, entre outros).
- O arquivo testa_eps é um teste de um caso extremo: a matriz não é nula, mas seus valores são menores que eps, então se comporta e é detectado como se fosse uma matriz nula.
- Os arquivos **ill_conditioned** e **ill_conditioned_perturbado** testam pequena perturbação em b e seu efeito em x.
- O teste do arquivo a_2x1 é interessante, pois foi escolhido propositalmente com dimensões pequenas (é um vetor de duas linhas) para ficar mais evidente o que está ocorrendo no programa. Mostra o caso de posto incompleto e como o QR calcula os mínimos quadrados. Num vetor que seja [1 1] transposto e b assuma os valores 5 e 9 (que é o que ocorre no arquivo), uma solução lógica esperada mentalmente seria 7, que é o que de fato acontece.
- Para compilar o arquivo que calcula os resíduos deve-se colocar:
  gcc -lm calcula_residuo.c -o calcula_residuo e para rodar:
  ./calcula_residuo <arquivo contendo A e b> <arquivo contendo x aproximado>
  em que x aproximado foi estimado a partir de um polinômio e que deseja-se calcular o resíduo para verificar se a aproximação foi boa.