

MAC 115 – Introdução à Computação – IAG

RAÍZES REAIS DE POLINÔMIOS ALEATÓRIOS

Segundo Exercício-Programa (EP2)

Entrega: **3 de junho de 2011**

A parte central deste EP consiste na implementação do método de Newton para encontrar raízes de polinômios. Você deverá escrever três programas, digamos, `ep2a.c`, `ep2b.c` e `ep2c.c`.

1. O seu programa `ep2a.c` deve receber como entrada um polinômio $p(x)$ e deve ter como saída uma (aproximação de uma) raiz de $p(x)$, ou uma mensagem dizendo que nenhuma raiz foi encontrada.
2. O seu programa `ep2b.c` deve receber como entrada um polinômio $p(x)$ e deve ter como saída a lista de todas as raízes reais de $p(x)$. (Naturalmente, novamente aqui queremos aproximações das raízes.)

Fazemos algumas considerações antes de descrever como deve ser seu `ep2c.c`. Suponha que

$$p(x) = a_0 + a_1x + \cdots + a_nx^n. \quad (1)$$

Para definirmos uma possível noção de ‘polinômio aleatório’ de grau n , fazemos o seguinte: supomos que $a_0 = 1$ e, para todo i ($1 \leq i \leq n$), fazemos a_i valer 1 ou -1 com igual probabilidade, independentemente. Em outras palavras, supomos que temos uma moeda honesta (isto é, que tem a mesma probabilidade de dar cara e de dar coroa) e a lançamos n vezes. Fazemos $a_i = 1$ se no i -ésimo lance obtivemos cara e fazemos $a_i = -1$ caso contrário. O polinômio $p = p(x)$ assim obtido é o *polinômio aleatório* em que estamos interessados. Queremos estudar o número $Z = Z(p)$ de raízes reais de p . O seu programa `ep2c.c` servirá para fazer um estudo experimental da quantidade Z . Seu programa deve estimar a média de Z :

3. O seu programa `ep2c.c` deve receber como entrada inteiros positivos n e M . Para estimar a média de Z , ele deve gerar M polinômios aleatórios $p(x)$ como em (1), seguindo o esquema acima. Para cada polinômio gerado p , seu programa deve determinar o número $Z(p)$ de raízes reais daquele polinômio. A saída de seu programa deve então ser a média dos $Z(p)$ assim obtidos. A saída de seu `ep2c.c` deve também incluir uma lista t_0, t_1, t_2, \dots , onde t_i é o número de polinômios encontrados com i raízes reais ($i = 0, 1, 2, \dots$). Para a conveniência do usuário, inclua também em sua saída a lista de frequências $t_0/M, t_1/M, t_2/M, \dots$; dessa forma, o usuário poderá saber qual é a proporção encontrada de polinômios sem raiz real, com uma raiz real, etc.

Números aleatórios. Para gerar os coeficientes aleatórios a_i ($1 \leq i \leq n$), você deve usar as funções `rand()` e `srand()` da biblioteca `stdlib`. Para o uso dessas funções, seu programa deve também receber como entrada uma ‘semente’ s . O uso de `rand()` e `srand()` será discutido em sala.

Bônus. Você poderá fazer uma versão alternativa do `ep2c.c` (bom para valores de n não muito grandes). Depois de escrever `ep2a.c`, `ep2b.c` e `ep2c.c`, você pode escrever `ep2bonus.c`, como segue:

4. O seu programa `ep2bonus.c` deve receber como entrada um inteiro positivo n . Ele deve gerar todos os 2^n polinômios $p(x)$ da forma (1) com $a_0 = 1$ e com $a_i \in \{-1, 1\}$ para todo $1 \leq i \leq n$.

Para cada polinômio $p = p(x)$ gerado, seu programa deve determinar o número de raízes $Z(p)$ de p . A saída de seu programa deve ser a média dos $Z(p)$ obtidos. A saída de `ep2bonus.c` deve também incluir uma lista t_0, t_1, t_2, \dots , onde t_i é o número de polinômios encontrados com i raízes reais ($i = 0, 1, 2, \dots$). Para a conveniência do usuário, inclua também em sua saída a lista de frequências $t_0 2^{-n}, t_1 2^{-n}, t_2 2^{-n}, \dots$, como em seu `ep2c.c`.

O método de Newton

Já conhecemos o método de Newton para encontrar aproximações de raízes das equações $f(x) = 0$. Começamos com um valor arbitrário x_0 e iteramos

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \quad (2)$$

para todo $k \geq 1$. Este processo define uma seqüência de reais x_0, x_1, x_2, \dots que converge (em geral) para uma raiz de $f(x) = 0$. No caso em que $f(x)$ é um polinômio, podemos determinar a derivada $f'(x) = p'(x)$ facilmente. Por exemplo, se $p(x)$ é como em (1), então

$$p'(x) = a_1 + a_2 x + \dots + n a_n x^{n-1}. \quad (3)$$

Representação de polinômios. Para representar o polinômio em (1) em seus programas, você pode usar um vetor $p[]$ com $p[i] = a_i$ ($0 \leq i \leq n$).

Detalhes de implementação. Na implementação do método de Newton, você pode usar a seguinte regra de parada: supondo dados um número real $\varepsilon > 0$ e um inteiro K , itere a recorrência (2) até que $|x_k - x_{k-1}| < \varepsilon$ ou $k = K$. Em seus programas, os parâmetros ε e K poderiam ser entradas, a serem fornecidas pelo usuário.

No `ep2a.c`, podemos usar um valor qualquer para x_0 ; possivelmente, o melhor é permitir ao usuário fornecer o valor de x_0 . Para listar todas as raízes de $p(x)$ no `ep2b.c`, você pode tentar vários valores de x_0 (valor inicial no método de Newton). Uma possibilidade seria experimentar os $T + 1$ valores

$$-A, -A + \frac{2A}{T}, -A + \frac{4A}{T}, \dots, A - \frac{4A}{T}, A - \frac{2A}{T}, A \quad (4)$$

para x_0 , onde $A > 0$ e um inteiro T são parâmetros dados. O usuário poderia fornecer A e T na entrada de seu `ep2b.c`.

No caso dos polinômios aleatórios definidos acima, é fácil provar que todas as suas raízes reais r satisfazem $1/2 < |r| < 2$.

OBSERVAÇÕES IMPORTANTES SOBRE OS EXERCÍCIOS-PROGRAMAS

Todos os exercícios-programas devem ter o seguinte cabeçalho:

```

/*****/
/* Fulano de Tal */
/* Exercício-Programa xx */
/* Curso yy - Turma zz -- Professor: Ciclano de Tal */
/* Computador: ..... */
/* Compilador: ..... */
/*****/

```

1. O exercício-programa é **estritamente individual**. Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO.

2. **Exercícios atrasados não serão aceitos.**
3. Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
4. É **muito importante** que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A avaliação dos exercícios-programas levará isto em conta.
5. Cada programa deve ser executado tantas vezes quantas forem necessárias para testar todos os casos possíveis para as entradas. Você pode dizer no seu programa (em comentários), quais dados você usou para testar seu programa. *A escolha desses dados também influirá na sua nota.*
6. Guarde com você até o final do semestre os arquivos de seus programas, pois eles podem ser solicitados pelo professor caso surja alguma dúvida.

IMPORTANTE. A entrega desse EP deve ser feita pelo Paca. O monitor dará dicas importantes sobre seu programa; fiquem atentos!