Terceiro Exercício-Programa

Norton Trevisan Roman
Fábio Nakano

14 de junho de 2012

1 Método de Newton-Raphson

O cálculo do "zero" de uma função consistem em, dada uma função $f(x) = 0$, achar o valor (ou possivelmente conjunto de valores) de x, para o qual essa expressão torna-se verdade. Por exemplo, considere a função $f(x) = x^2 + 4$. Um zero dessa função seria:

$$\begin{array}{rcccl} f(x) & = & x^2 - 4 & = & 0 \\ & & x^2 & = & 4 \\ & & x & = & 2 \end{array}$$

sendo que o outro zero estaria em $x = -2$.

Esse mesmo raciocínio pode ser usado também para se resolver equações. Por exemplo, considere a equação $x^2 = 4$. Ela pode ser reduzida ao problema de se encontrar o zero de $F(x) = x^2 - 4$.

Naturalmente, nesse exemplo a equação é bastante simples e direta. Contudo, o poder do método se mostra em equações mais complexas. Por exemplo, imagine que você faz uma determinada aplicação inicial D_0 em um fundo de investimento, em uma determinada data t_0. A partir dessa data, e em dias esporádicos, você faz novos depósitos D_i nesse mesmo fundo. Passado um certo tempo, você verifica o saldo do fundo, e então se pergunta qual teriam sido os juros (compostos) médios mensais praticados no período?

Essa resposta seria razoavelmente simples, caso os depósitos tivessem sido constantes. Contudo, eles foram esporádicos: você somente depositou quando o acaso fez sobrar algum dinheiro.

Como fazer então? Imagine que foram feitos n depósitos (além do inicial, ou seja, $n + 1$ no total). Para simplificar considere que todo depósito, quando feito, é feito no dia primeiro do mês.

depósito	*data*
D_0	t_0
D_1	t_1
...	...
D_n	t_n

Em uma determinada data t_f, você verifica o saldo S. Supondo uma taxa mensal de juros $0 \leq j \leq 1$ constante, podemos aplicar esses juros a cada um dos depósitos, resultando em

$$D_0(1+j)^{t_f-t_0} + D_1(1+j)^{t_f-t_1} + \ldots + D_n(1+j)^{t_f-t_n} = S$$

Qual seria então a taxa de juros? Note o problema é, na verdade, encontrar o zero da função

$$f(j) = D_0(1+j)^{t_f-t_0} + D_1(1+j)^{t_f-t_1} + \ldots + D_n(1+j)^{t_f-t_n} - S$$

E como funciona o método de Newton-Raphson para zeros de funções? O método de Newton-Raphson parte de uma aproximação inicial para a variável buscada (um palpite para j; 0.5 por exemplo), e incrementalmente vai chegando cada vez mais perto da solução. Formalmente, o método diz que[1]:

$$\begin{aligned} j_0 &\leftarrow \frac{1}{2} \\ j_{k+1} &\leftarrow j_k - \frac{f(j_k)}{f'(j_k)} \end{aligned}$$

para $k = 0, 1, 2, \ldots$. Ou seja, obtemos j_1 fazendo $j_0 - \dfrac{f(j_0)}{f'(j_0)}$, e assim por diante. Note que quanto mais alto o valor de k, melhor a aproximação de $f(j)$ dada por j_{k+1}. Nessa equação, $f'(j_k)$ é a derivada de $f(j)$ no ponto j_k. Como isso vocês verão mais tarde no curso de Cálculo I, podemos adiantar que a derivada de

$$f(j) = D_0(1+j)^{t_f-t_0} + D_1(1+j)^{t_f-t_1} + \ldots + D_n(1+j)^{t_f-t_n} - S$$

é

$$f'(j) = (t_f - t_0)D_0(1+j)^{t_f-t_0-1} + (t_f - t_1)D_1(1+j)^{t_f-t_1-1} + \ldots + (t_f - t_n)D_n(1+j)^{t_f-t_n-1}$$

Esse processo deve ser repetido enquanto $|j_{k+1} - j_k| \geq \epsilon$, onde ϵ é um número positivo que representa a precisão do cálculo. Assim, a aproximação de $f(j)$ será o primeiro valor j_{k+1} para o qual $|j_{k+1} - j_k| < \epsilon$.

Por exemplo, suponha que sejam feitos 5 depósitos, conforme a tabela abaixo:

[1] Para um método mais geométrico de se obter essa equação, veja http://www.youtube.com/watch?v=6ueocPki25I . Para uma dedução com base na série de Taylor, consulte http://omnis.if.ufrj.br/ sandra/MetComp/2011-1/Newton_Raphson.pdf .

Valor	*Data*
1.000,00	03/2011
1.200,00	04/2011
100,00	05/2011
1.100,00	07/2011
900,00	09/2011

Suponha agora que, em 12/2011, o saldo seja de 5.000,00. Assim, $f(j)$, com cada D_i dado pela tabela acima, $t_f = 12/2011$ e $S = 5.000,00$, será

$$\begin{aligned} f(j) &= 1.000,00(1+j)^9 + 1.200,00(1+j)^8 + 100,00(1+j)^7 + 1.100,00(1+j)^5 \\ &\quad +900,00(1+j)^3 - 5.000,00 \end{aligned}$$

e

$$\begin{aligned} f'(j) &= 9 \cdot 1.000,00(1+j)^8 + 8 \cdot 1.200,00(1+j)^7 + 7 \cdot 100,00(1+j)^6 \\ &\quad +5 \cdot 1.100,00(1+j)^4 + 3 \cdot 900,00(1+j)^2 \end{aligned}$$

Segundo o método de Newton-Raphson, $f(j)$, com $\epsilon = 0.001$ fica:

iteração	j_k	j_{k+1}	$\lvert j_{k+1} - j_k \rvert$
1	0.5	0.3229471438873768	0.17705285611262322
2	0.3229471438873768	0.17719505698725935	0.14575208690011743
3	0.17719505698725935	0.07603810202653498	0.10115695496072437
4	0.07603810202653498	0.031151323858092876	0.0448867781684421
5	0.031151323858092876	0.023807799064652573	0.007343524793440304
6	0.023807799064652573	0.023637054483966326	1.7074458068624607E-4

e a resposta será 0.023637054483966326. Ou seja, para que essa série de depósitos tenha gerado o montante de 5.000,00, é necessário que os juros mensais tenham sido de 0.023637054483966326 (ou $\approx 2,36\%$) ao mês.

2 Tarefa

Você deve completar o método `public static double juros(ListaDepositos depositos, Deposito saldo, double epsilon)`, da classe `Juros`. Esse método recebe como parâmetros o valor de ϵ, uma lista de depósitos (objetos de classe que implemente a interface `lista.ListaDepositos` – já há uma implementação no pacote `lista`), e um saldo final (objeto da classe `depositos.Deposito`), retornando $f(j)$ acima descrita.

Note que, ao contrário do EP_1, não há limite para o número de depósitos, ou seja, a lista pode conter um número arbitrário de elementos. Além disso, a lista não pode estar vazia (ou ser nula), o saldo final também não pode ser nulo, e $0 < \epsilon < 1$.

ATENÇÃO! À exceção da classe `Juros`, você não pode modificar nenhuma outra classe. Para os testes, será considerada tão somente o arquivo `Juros.java` entregue.

2.1 Entrada

A entrada é composta pela lista de Depósitos, o saldo final, além da precisão $0 < \epsilon < 1$.

2.2 Saída

Como saída, o método retorna o valor de j com precisão ϵ, ou `Double.NaN`, caso $\epsilon \leq 0$, $\epsilon \geq 1$, a lista ou o saldo forem `null`, ou a lista estiver vazia.

2.3 Material a Ser Entregue

Deverá ser entregue tão somente o arquivo Juros.java. No início do arquivo, acrescente um cabeçalho informativo, como o seguinte:

```
/**********************************************************************/
/**   ACH2001 - Introdução à Ciência da Computação I                 **/
/**   EACH-USP - Primeiro Semestre de 2011                           **/
/**   <turma> - <nome do professor>                                  **/
/**                                                                  **/
/**   Terceiro Exercício-Programa                                    **/
/**   Arquivo: <nome do arquivo>                                     **/
/**                                                                  **/
/**   <nome do(a) aluno(a)>                   <número USP>           **/
/**                                                                  **/
/**   <data de entrega>                                              **/
/**********************************************************************/
```

A entrega será feita unica e exclusivamente via col, até a data marcada para entrega. Deverá ser postado no col um zip ou rar com o arquivo Juros.java. O nome do arquivo compactado deve ser seu número USP, ou seja:

```
númeroUsp.zip
```

Somente este arquivo zip deve ser postado no col. A responsabilidade de postagem nele é exclusiva do aluno. Por isso, problema referentes ao uso do sistema devem ser resolvidos com antecedência.

3 Avaliação

O programa entregue será avaliado conforme sua corretude. Além disso, algumas observações pertinentes ao trabalho, que influem em sua nota, são:

- Este exercício-programa deve ser elaborado individualmente.
- Não será tolerado plágio, em hipótese alguma.
- Exercícios com erro de sintaxe (ou seja, erros de compilação), receberão nota ZERO

Atenção! Para avaliação, apenas o método `juros(ListaDepositos depositos, Deposito saldo, double epsilon)` será invocado diretamente. Em especial, qualquer código dentro do *main* será ignorado. Então tenha certeza de que o problema é resolvido chamando-se diretamente somente esse método.