

MS211 - Cursão - 2007

Quarto Exercício Programa

Um Tributo a Leonhard Euler: Números Complexos, Conjuntos Fractais e a Hipótese de Riemann

Resumo

Ao contrário do FORTRAN, a linguagem C não possui variáveis do tipo complexo. Neste EP, veremos que esta deficiência pode ser facilmente sanada usando-se certas estruturas de dados. Como aplicação, deverão ser feitos programas geradores de conjuntos fractais de Julia e de Mandelbrot, assim como a implementação da função zeta de Riemann e a introdução da célebre Hipótese de Riemann.

Leonhard Euler contribuiu expressivamente à todos os pontos discutidos neste EP. Gostaria que este trabalho fosse encarado como uma homenagem a Euler, na véspera do seu tricentésimo aniversário (15/04).

1 Variáveis complexas em C

A linguagem C não possui um tipo de dado para variáveis complexas, ao contrário do FORTRAN¹. Varias aplicações científicas, porém, requerem o uso de variáveis complexas. A flexibilidade da linguagem C permite que esta deficiência seja facilmente sanada usando-se as definições do tipo `structure`. Por exemplo,

```
struct complex
{
    double r; /*parte real*/
    double i; /*parte imaginaria*/
};
```

¹e do C++, mais informações, aqui.

define a estrutura chamada `complex`, formada por dois reais de precisão dupla. Uma variável complexa `a` pode ser introduzida através da estrutura

```
struct complex a;
```

Suas duas componentes, sua parte real e sua parte imaginária, podem ser invocadas usando-se, respectivamente, `a.r` e `a.i`. Pode-se, também, definir funções cujas saídas são números complexos, como no exemplo

```
struct complex conjug(struct complex a)
{
    struct complex b;
    b.r = a.r;
    b.i = -a.i;
    return(b);
}
```

Como pode-se ver, trata-se de uma função que, dado um número complexo `a`, retorna seu complexo conjugado. As partes reais e imaginárias das variáveis complexas podem ser invocadas da maneira usual, como números reais, como exemplificado no programa a seguir, disponível em nosso repositório,

```
/******  
/*    complex.c    ASaa@20070405    */  
/******  
#include <stdio.h>  
#include <stdlib.h>  
  
struct complex  
{  
    double r; /*parte real*/  
    double i; /*parte imaginaria*/  
};  
  
struct complex conjug(struct complex a)  
{  
    struct complex b;  
    b.r = a.r;  
    b.i = -a.i;  
    return(b);  
}
```

```

}

int main(){
struct complex a,b;
    printf("Parte real: ");
    scanf("%lf", &a.r ) ;

    printf("Parte imaginaria: ");
    scanf("%lf", &a.i ) ;

    b = conjug(a);
    printf("Complexo conjugado: %lf %lf \n ",b.r,b.i) ;
}

```

Os dois itens deste EP que serão expostos a seguir exigirão uma série de funções para manipulação de números complexos além da função *conjug* citada acima. Dentre elas, destacam-se:

`struct complex soma(struct complex a , struct complex b)` : Retorna a soma dos números complexos a e b.

`struct complex subtrai(struct complex a , struct complex b)` : Retorna a diferença dos números complexos a e b.

`struct complex mult(struct complex a , struct complex b)` : Retorna a multiplicação dos números complexos a e b. Lembrem-se que $ab = (a_r + ia_i)(b_r + ib_i) = a_rb_r - a_ib_i + i(a_rb_i + a_ib_r)$.

`struct complex divis(struct complex a , struct complex b)` : Retorna a divisão dos números complexos a e b. Lembrem-se que $a/b = (a\bar{b})/(b\bar{b}) = (a_r + ia_i)(b_r - ib_i)/(b_r^2 + b_i^2)$.

`struct complex potenc(struct complex a , integer n)` : Retorna a^n , *i.e.*, o número complexo a elevado a potência n.

`double modulo(struct complex a)` : Retorna o módulo do número complexo a.

`double argumento(struct complex a)` : Retorna o argumento (ângulo na forma polar) do número complexo a.

Como aquecimento, sugere-se a implementação de todas estas funções para testes. Não é necessário enviar estes testes.

2 Conjuntos fractais de Julia e de Mandelbrot

Os conjuntos fractais de Julia e de Mandelbrot são obtidos do estudo do comportamento assintótico da seguinte seqüência de números complexos

$$z_{n+1} = z_n^2 + c. \quad (1)$$

O conjunto de Julia J_c é definido fixando-se uma constante $c \in \mathbb{C}$

$$J_c = \left\{ z_0 \in \mathbb{C} \mid \lim_{n \rightarrow \infty} |z_n| < \infty \right\}.$$

Em outras palavras, o conjunto de Julia J_c corresponde aos números complexos z_0 , os quais, usados como condição inicial de (1), dão origem a séries que não divergem para o infinito, *i.e.*, séries que permanecem para grandes n numa vizinhança da origem. Já o conjunto de Mandelbrot M_2 é definido a partir da seqüência (1) como

$$M_2 = \left\{ c \in \mathbb{C} \mid \lim_{n \rightarrow \infty} |z_n| < \infty, \text{ com } z_0 = 0 \right\}, \quad (2)$$

i.e., corresponde ao conjunto de números complexos c tais que a seqüência (1), iterada a partir de $z_0 = 0$, não diverge para o infinito. Como vemos, M_2 e J_c são conjuntos relacionados. São muito bem conhecidos também, vejam alguns exemplos nas Figs. 1 e 2.

O primeiro item deste EP consiste na elaboração de um programa capaz de gerar graficamente, usando-se arquivos ppm ou pgm, conjuntos de Julia e de Mandelbrot. No caso dos conjuntos de Julia, o usuário deve fornecer o valor de c (pode-se admitir apenas $|c| < 2$) e o programa deve gerar a imagem gráfica de J_c explorando-se, por exemplo, um conjunto de 400×400 pontos correspondentes a região $|\operatorname{Re}(z)| \leq 5/2$ e $|\operatorname{Im}(z)| \leq 5/2$ do plano complexo \mathbb{C} . Use, inicialmente, um máximo de 100 iterações da seqüência (1) para verificar se $\{z_n\}$ diverge. Para discussão nas aulas exploratórias, deverão ser respondidas as seguintes perguntas:

1. Qual o menor valor de $|z_n|$ que pode ser usado para identificar as seqüências tais que $\lim_{n \rightarrow \infty} |z_n| \rightarrow \infty$?

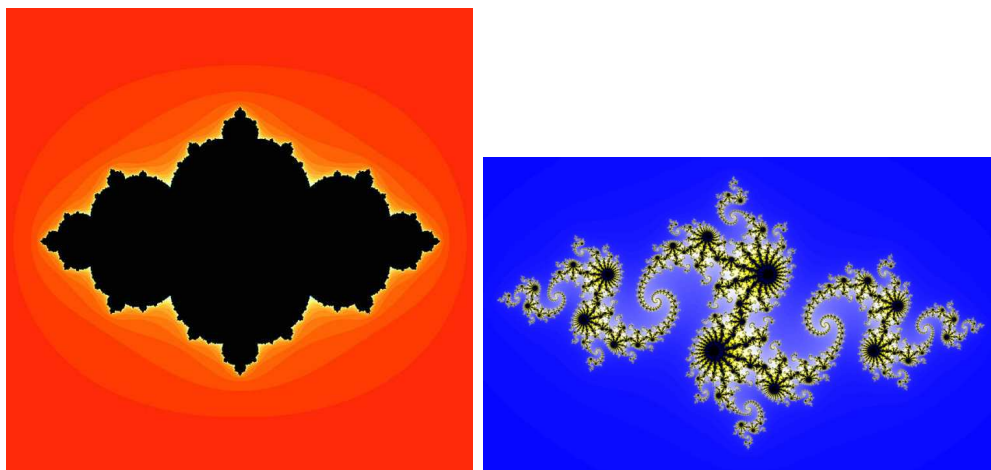


Figura 1: Conjuntos de Julia J_c . Esquerda: Os pontos pintados de preto correspondem ao conjunto de Julia obtido para $c = \varphi - 2$, sendo $\varphi = (1 + \sqrt{5})/2$ a razão áurea. Direita: Idem, para o caso $c = -0.8 + 0.156i$. Discutiremos o significado das cores nestes gráficos nas aulas exploratórias. (Imagens retiradas do artigo correspondente da Wikipedia.)

2. Para os casos em que $\{z_n\}$ não diverge, qual seu comportamento para n grandes?
3. Como se alteram os conjuntos escolhendo-se outros valores (menores e maiores) para o número máximo de iterações?

Para o caso dos conjuntos de Mandelbrot, o programa deve gerar uma imagem gráfica de M_n , definido da mesma maneira que (2), mas a partir da seqüência

$$z_{k+1} = z_k^n + c.$$

Utilize as mesmas definições usadas para o caso de J_c e responda, para as aulas exploratórias, as mesmas perguntas.

3 A função zeta da Riemann

A função zeta de Riemann $\zeta(z)$ é definida pela série

$$\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}, \quad (3)$$

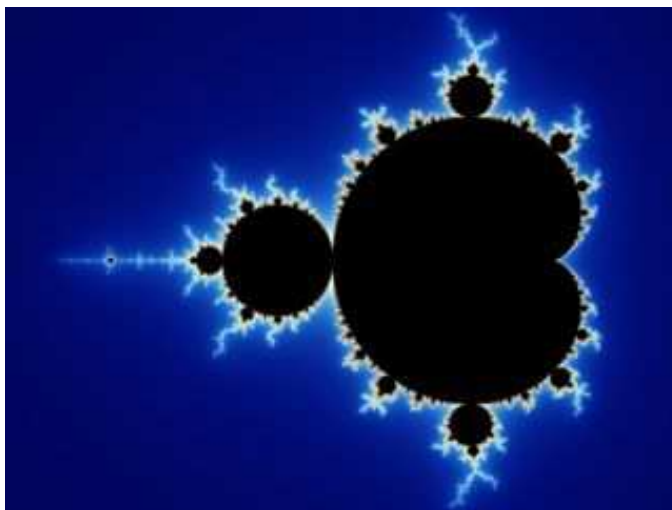


Figura 2: Os pontos pintados de preto pertencem ao conjunto de Mandelbrot M_2 . Discutiremos o significado das cores neste gráfico nas aulas exploratórias. (Imagem retirada do artigo correspondente da Wikipedia.)

para todo $z \in \mathbb{C}$ com $\text{Re}(z) > 1$. De fato, a definição da função $\zeta(z)$ pode ser estendida, por meio de uma *continuação analítica*², para todo o plano complexo, excluindo-se apenas o ponto $z = 1$, para o qual a série diverge ($\zeta(1)$ é a série harmônica $1 + 1/2 + 1/3 + 1/4 + \dots$, cuja divergência também foi estabelecida por Euler). Não é de se estranhar que Euler já conhecesse a função $\zeta(z)$ com anos antes de Riemann, e que soubesse também de sua profunda relação com os números primos. Vamos nos restringir, momentaneamente, a argumentos reais ($\text{Im}(z) = 0$, $\text{Re}(z) = x$)

$$\zeta(x) = 1 + \frac{1}{2^x} + \frac{1}{3^x} + \frac{1}{4^x} + \frac{1}{5^x} + \dots$$

Multiplicando-se ambos os lados por $1/2^x$ e subtraindo-se o resultado da expressão original, obtém-se

$$\left(1 - \frac{1}{2^x}\right) \zeta(x) = 1 + \frac{1}{3^x} + \frac{1}{5^x} + \frac{1}{7^x} + \frac{1}{9^x} + \dots$$

Constata-se que todos os termos envolvendo múltiplos de 2 desapareceram do lado direito da igualdade. Repetindo-se o mesmo procedimento, multiplicando-

²Veja, por exemplo, R.V. Churchill, *Complex Variables and Applications*.

se agora por $1/3^x$, obtem-se

$$\left(1 - \frac{1}{3^x}\right) \left(1 - \frac{1}{2^x}\right) \zeta(x) = 1 + \frac{1}{5^x} + \frac{1}{7^x} + \frac{1}{11^x} + \frac{1}{13^x} + \dots$$

Nota-se, agora, que não restam múltiplos de 3 nem de 2 do lado direito. Continuando-se este processo³ indefinidamente com todos os outros primos, obtem-se finalmente

$$\dots \left(1 - \frac{1}{7^x}\right) \left(1 - \frac{1}{5^x}\right) \left(1 - \frac{1}{3^x}\right) \left(1 - \frac{1}{2^x}\right) \zeta(x) = 1,$$

ou, de uma maneira mais concisa,

$$\zeta(x) = \prod_{\text{primos } p} (1 - p^{-x})^{-1},$$

resultado obtido pela primeira vez por Euler⁴.

A chamada Hipótese de Riemann é uma das grandes conjecturas da Matemática, listada pelo Clay Mathematics Institute como um dos problemas do Milênio, sendo merecedor, quem a resolver, de um prêmio de US\$ 1.000.000,00. Formulada como uma conjectura por Bernhard Riemann em 1859 no seu trabalho *Sobre o número de primos menores que uma certa magnitude*⁵, ela afirma que todas as raízes não triviais da função $\zeta(z)$ estão sobre a linha reta correspondente a $\text{Re}(z) = 1/2$. Diz-se raízes não triviais para diferenciá-las de outras raízes conhecidas da continuação analítica de (3) correspondentes aos casos de z reais, inteiros e negativos. A intenção deste segundo item do EP é implementar a função $\zeta(z)$ em linguagem C e testar a Hipótese de Riemann.

Pode-se usar a definição (3) diretamente para calcular-se a função $\zeta(z)$ para um dado z específico com $\text{Re}(z) > 1$. O grande problema do cálculo de qualquer função definida a partir de uma série sempre é: como estimar o número de termos a serem somados para obter-se o valor de $\zeta(z)$ com uma precisão ε pré-estabelecida? Em outras palavras, como estimar o erro

$$\xi_N = \zeta(z) - \sum_{k=1}^N \frac{1}{k^z} \tag{4}$$

³Semelhante à peneira de Eratóstenes.

⁴Este resultado é válido também para argumentos complexos com $\text{Re}(z) > 1$.

⁵ *Über die Anzahl der Primzahlen unter einer gegebenen Größe*. Para uma tradução inglesa do trabalho original, clique aqui.

cometido truncando-se a série em $k = N$? Antes de continuarmos esta discussão, convém relembrar o significado de k^{-z} , sendo k um inteiro e z um complexo:

$$k^{-z} = e^{-z \ln k} = e^{(-\operatorname{Re}(z) - i\operatorname{Im}(z)) \ln k} = k^{-\operatorname{Re}(z)} (\cos(\operatorname{Im}(z) \ln k) - i \sin(\operatorname{Im}(z) \ln k)). \quad (5)$$

Como vemos, k^{-z} é um número complexo para o qual $|k^{-z}| = k^{-\operatorname{Re}(z)}$. O erro (4) pode então ser estimado como

$$|\xi_N| = \left| \sum_{k=N+1}^{\infty} \frac{1}{k^z} \right| \leq \sum_{k=N+1}^{\infty} |k^{-z}| = \sum_{k=N+1}^{\infty} k^{-\operatorname{Re}(z)}. \quad (6)$$

Um resultado de Euler é essencial para a estimativa deste erro. Trata-se do chamado problema da Basileia, sua cidade natal. O problema consiste em calcular-se a soma dos recíprocos dos quadrados dos números inteiros:

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$$

Como podemos ver, este é o valor de $\zeta(2)$. Conhecendo-o, podemos inferir, para $\operatorname{Re}(z) \geq 2$, um valor máximo para o erro ξ_N

$$|\xi_N| \leq \sum_{k=N+1}^{\infty} k^{-\operatorname{Re}(z)} \leq \sum_{k=N+1}^{\infty} k^{-2} = \epsilon_N. \quad (7)$$

Euler resolveu o problema da Basileia da seguinte maneira. Considere a série de Taylor⁶ em torno de $x = 0$ da função $(\sin x)/x$

$$\frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} \dots \quad (8)$$

As raízes de $(\sin x)/x$ estão localizadas nos pontos $x = \pm k\pi$, $k = 1, 2, 3, \dots$. Portanto, fatorando-se o polinômio (infinito) do lado direito de (8), deve-se obter

$$\begin{aligned} \frac{\sin x}{x} &= \left(1 - \frac{x}{\pi}\right) \left(1 + \frac{x}{\pi}\right) \left(1 - \frac{x}{2\pi}\right) \left(1 + \frac{x}{2\pi}\right) \left(1 - \frac{x}{3\pi}\right) \left(1 + \frac{x}{3\pi}\right) \dots \\ &\quad \left(1 - \frac{x^2}{\pi^2}\right) \left(1 - \frac{x^2}{4\pi^2}\right) \left(1 - \frac{x^2}{9\pi^2}\right) \left(1 - \frac{x^2}{16\pi^2}\right) \dots \end{aligned}$$

⁶ Isto pode ser calculado facilmente a partir da série de Taylor da função $\sin x$, dividindo-se a série por x .

N	ϵ_N
10	< 0.1
100	< 0.01
1000	< 0.001
10000	< 0.0001

Tabela 1: Erro ϵ_N cometido ao se aproximar $\zeta(2)$ pela soma dos N primeiros termos da série (3) para $z = 2$. Nota-se claramente que $\epsilon_N < 1/N$.

Expandindo-se este último polinômio, vemos que o coeficiente do termo x^2 será

$$-\frac{1}{\pi^2} \left(1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots \right)$$

Porém, já sabemos de (8) que este coeficiente deve ser $-1/3! = -1/6$, finalmente estabelecendo que

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots = \frac{\pi^2}{6} = \zeta(2).$$

Este resultado permite-nos avaliar ϵ_N de (7) comparando-se o resultado exato de $\zeta(2)$ com o valor obtido somando-se os apenas os N primeiros termos da série. Note ainda que ϵ_N corresponde ao erro *máximo* cometido para $\text{Re}(z) \geq 2$. Os resultados estão na Tabela 1. Como vemos, a situação está sob controle para $\text{Re}(z) \geq 2$.

Porém, para verificarmos a Hipótese de Riemann, devemos calcular $\zeta(z)$ para $z = 1/2 + it$, $t \in \mathbb{R}$. Nesta região, sequer a expressão (3) é válida. Novamente, recorreremos aos trabalhos de Euler, em particular, àqueles sobre aceleração de convergência de séries. Euler mostrou que, associadas a séries como (3), sempre há séries alternantes com mesmo limite que convergem mais rapidamente. Muitas vezes, muito mais rapidamente. Para este fim, consideremos a seguinte série alternante:

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{n^z} = -1 + \frac{1}{2^z} - \frac{1}{3^z} + \frac{1}{4^z} - \frac{1}{5^z} + \frac{1}{6^z} + \dots$$

Somando-se, termo a termo, com a série (3), obtém-se

$$2 \left(\frac{1}{2^z} + \frac{1}{4^z} + \frac{1}{6^z} + \frac{1}{8^z} + \frac{1}{10^z} \dots \right) =$$

N	$ \epsilon_N $
10	< 0.37
100	< 0.12
1000	< 0.038
10000	< 0.012

Tabela 2: Erro ϵ_N cometido ao se aproximar $\zeta(1/2)$ pela soma dos N primeiros termos da série (9) para $z = 1/2$, de maneira análoga ao mostrado na Tabela 1. Nota-se claramente que $\epsilon_N \approx 1/\sqrt{N}$, uma taxa muito mais lenta que as obtidas no caso de $\zeta(2)$.

$$\frac{2}{2^z} \left(1 + \frac{1}{2^z} + \frac{1}{3^z} + \frac{1}{4^z} + \frac{1}{5^z} + \dots \right) = 2^{1-z} \zeta(z).$$

Finalmente, obtém-se

$$\zeta(z) = \frac{1}{1 - 2^{1-z}} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^z}. \quad (9)$$

A primeira grande vantagem de (9) é que sua convergência é muito mais rápida do que a de (3). Pode-se mostrar, por exemplo, que o erro ϵ_N cometido aproximando-se $\zeta(2)$ pela soma dos N primeiros termos de (9) é tal que $\epsilon_N < 1/N^2$, compare com os valores da Tabela 1. Porém, nossa principal vantagem é que (9) converge para todo $z \in \mathbb{C}$ com $\text{Re}(z) > 0$! Podemos usar, portanto, a série (9) na região onde a Hipótese de Riemann deve ser testada. A convergência, porém, é muito mais lenta (e muitíssimo mais difícil de se estimar...). A Tabela 2 apresenta alguns valores para o erro ϵ_N envolvido no cálculo de $\zeta(1/2)$, obtido a partir de técnicas fora de nosso escopo atual⁷. Notem que, como no caso anterior, ϵ_N será o máximo erro cometido para $\text{Re}(z) \geq 1/2$.

Já estamos em condições de enunciar o segundo item do EP. Trata-se da implementação da função

```
struct complex zeta(struct complex z, double eps)
```

a qual deve retornar o valor de $\zeta(z)$ com erro inferior a eps , supondo-se $\text{Re}(z) \geq 1/2$ e $z \neq 1$. Para o caso em que $\text{Re}(z) \geq 2$, *deve-se* usar a série (3)

⁷Vejam, por exemplo, Milton Abramowitz and Irene A. Stegun, *Handbook of Mathematical Functions*.

e os valores da Tabela 1. Já para $1/2 \leq \text{Re}(z) < 2$, deve-se usar a série (9) e os valores da Tabela 2. A implementação *deve* também usar a função

```
struct complex kz(integer k, struct complex z)
```

a qual retorna o número complexo k^z , nos moldes de (5).

Uma vez implementada a função $\zeta(z)$, podemos fazer diversos testes da Hipótese de Riemann. Pode-se, por exemplo, calcular-se $\zeta(1/2 + it)$, $t \in [0, 40]$, e verificar a presença de zeros nesse intervalo. De maneira análoga, pode-se verificar a ausência de zeros para qualquer intervalo envolvendo z tal que $\text{Re}(z) > 1/2$. A melhor maneira de visualizar-se estes resultados é fazendo-se gráficos no plano complexo, como o mostrado na Fig. 3. Para gerar-se o gráfico, pode-se usar qualquer pacote. Sugere-se

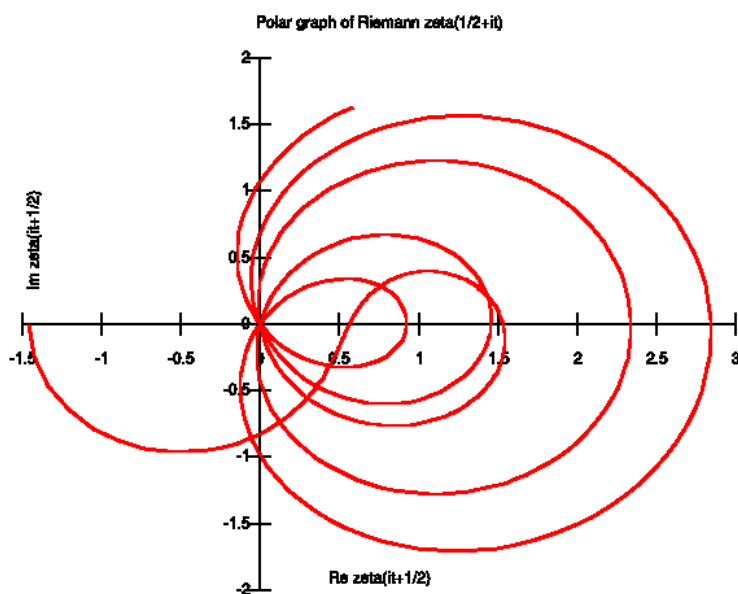


Figura 3: Curva parametrizada correspondente a $\zeta(1/2 + it)$. Seus 5 primeiros zeros são evidentes nos pontos de cruzamento com a origem. (*Imagem retirada do artigo correspondente da Wikipedia.*)

o `gnuplot`. Mais informações, nas aulas exploratórias, onde também discutiremos alguns métodos, quase todos propostos por Euler, para se acelerar a convergência de séries como as envolvidas no cálculo da função $\zeta(z)$.

Agradecimentos

Pedro Henrique, David e Bowie.