Métodos Computacionais
Trabalho I

Daniel De Luca, Diego Jornada, Matthias Nunes

[1]Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

{daniel.luca, diego.jornada, matthias.nunes}@acad.pucrs.br

Resumo. *Este artigo descreve uma síntese sobre computação científica e alguns métodos iterativos, quando utilizados para achar números irracionais.*

1. Introdução

Computação científica é a área que estuda a construção de modelos matemáticos e técnicas para análises quantitativas para resolver problemas cinetíficos e de engenharia.

2. Métodos Iterativos

Um **método iterativo** é um procedimento matemático para resolução de equações que gera uma sequência de aproximações, cada vez melhores, que servem como solução para uma classe de problemas. Uma implementação específica de um método iterativo, incluindo sua condição de parada, é um algoritmo. Um método iterativo é considerado **convergente** se a sequência de valores geradas converge para uma dada aproximação inicial.

Para uma definição mais teórica, o seguinte autor define:

> Um método iterativo consiste em repetir uma determinada operação um certo número de vezes até que nos seja fornecida uma aproximação, que satisfaça as condições do problema e, para tal, a sequência de valores deve ser convergente.[Batista 2014]

Nos problemas do tipo *encontre a raíz da equação*, um método iterativo usa uma suposição inicial para gerar sucessivas aproximações à uma solução. Em Contraste, **métodos diretos** tentam resolver o problema em uma sequência *finita* de operações.

Um método iterativo é formado por quatro partes: [Cláudio 2000]

a) Estimativa inicial: uma ou mais aproximações para a raiz desejada.
b) Atualização: uma fórmula que atualize a solução aproximada.
c) Critério de parada: uma forma de estabelecer quando parar o processo iterativo em qualquer caso.
d) Estimador de exatidão: está associado ao critério de parada e provê uma estimativa do erro cometido.

3. Exercícios

Nessa seção vamos encontrar números irracionais conhecidos utilizando métodos iterativos.

3.1. Número de Ouro

3.1.1. Frações Continuadas

Para encontrarmos o número de ouro utilizamos o ambiente matemático *octave*, e conseguimos gerar a seguinte tabela:

Iteração	x	Iteração	x
1	2.00000000000000000000	26	1.61803398873830306393
2	1.50000000000000000000	27	1.61803398875432247195
3	1.66666666666666651864	28	1.61803398874820381081
4	1.60000000000000008882	29	1.61803398875054060824
5	1.62500000000000000000	30	1.61803398874964821097
6	1.61538461538461541878	31	1.61803398874998904944
7	1.61904761904761906877	32	1.61803398874985893130
8	1.61764705882352943789	33	1.61803398874990866929
9	1.61818181818181816567	34	1.61803398874988957346
10	1.61797752808988759554	35	1.61803398874989690093
11	1.61805555555555558023	36	1.61803398874989401435
12	1.61802575107296142676	37	1.61803398874989512457
13	1.61803713527851456000	38	1.61803398874989490253
14	1.61803278688524576623	39	1.61803398874989490253
15	1.61803444782168193150	40	1.61803398874989490253
16	1.61803381340012508716	41	1.61803398874989490253
17	1.61803405572755432118	42	1.61803398874989490253
18	1.61803396316670644595	43	1.61803398874989490253
19	1.61803399852180351814	44	1.61803398874989490253
20	1.61803398501735795634	45	1.61803398874989490253
21	1.61803399017559712547	46	1.61803398874989490253
22	1.61803398820532517988	47	1.61803398874989490253
23	1.61803398895790184753	48	1.61803398874989490253
24	1.61803398867044334608	49	1.61803398874989490253
25	1.61803398878024262686	50	1.61803398874989490253

Tabela 1. Convergência do número de ouro pelo método de frações continuadas

Como fica ilustrado na tabela, o número já se estabiliza antes mesmo de chegarmos à 50 iterações.

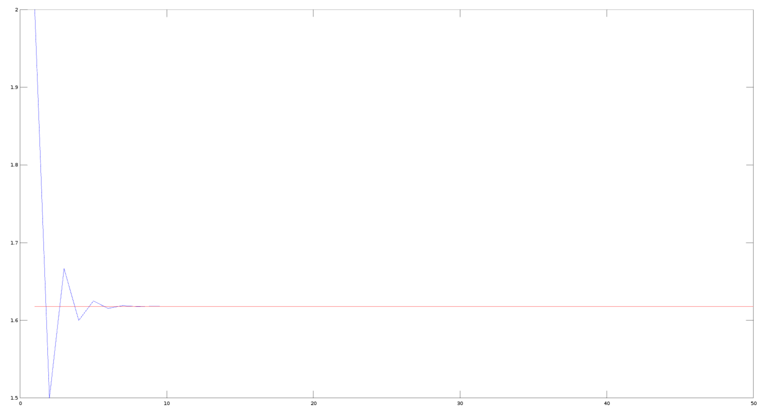

Figura 1. Convergência do número de ouro pelo método de frações continuadas

O método iterativo utilizado foi descrito pelo seguinte código:

```
function [x] = phi_frac(iteration=1)
  aux = 1;
  for i= 1:iteration
    aux = double(1 + 1/aux);
    x(i) = aux;
  end
end
```

3.1.2. Método de Newton

A ideia do método de Newton é a seguinte: É estabelecido um chute inicial, que seja razoavelmente perto da raíz verdadeira, então a função é aproximada por sua reta tangente. O x que intercepta a reta e a função é computado, e ele será uma melhor aproximação que o chute inicial. O método, então, pode ser iterado.

Analizando novamente o número de ouro, mas com o método de newton, um número muito menor de iterações é observado.

Iteração	x	erro
1	1.66666666666666674068	0.33333333333333325932
2	1.61904761904761906877	0.04761904761904767192
3	1.61803444782168193150	0.00101317122593713727
4	1.61803398874998904944	0.00000045907169288206
5	1.61803398874989490253	0.00000000000009414691
6	1.61803398874989490253	0.00000000000000000000

Tabela 2. Convergência do número de ouro pelo método de newton

Isso pode ser observado no gráfico abaixo.

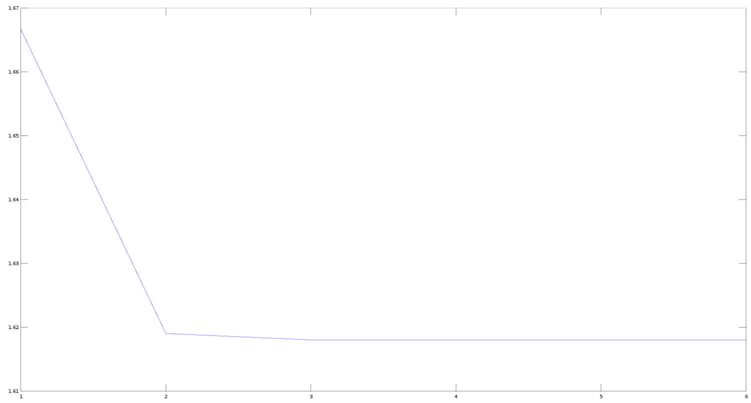

Figura 2. Convergência do número de ouro pelo método de newton

O método iterativo utilizado foi descrito pelo seguinte código:

```
function [x, ex] = newton( f, df, x0, tol, nmax)
  f = inline(f);
  df = inline(df);
  x(1) = double(x0 - (f(x0)/df(x0)));
  ex(1) = abs(x(1)-x0);
  k = 2;
  while k <= nmax && ex(k-1) > tol
     x(k) = double(double(x(k-1)) - double((f(x(k-1))/df(x(k-1)))));
     ex(k) = abs(x(k)-x(k-1));
     k = k+1;
  end
end
```

3.2. Pi(π)

3.2.1. Método Misterioso

Encontramos em um fórum de matemática um método iterativo que calcula π de uma forma aparentemente mais simples, apesar de sua complexidade estar escondida na função sin. O método está descrito a seguir:

$$P(n+1) = P(n) + sin(P(n)) \tag{1}$$

$P(n)$ seria a aproximação de π na iteração n. Esse método consegue convergir para π com um número muito baixo de iterações.

$P(n)$
3.14112000805986735230135309393517673015594482421875
3.14159265357219563696844488731585443019866943359375
3.14159265358979311599796346854418516159057617187500
3.14159265358979311599796346854418516159057617187500
3.14159265358979311599796346854418516159057617187500

π
3.14159265358979311599796346854418516159057617187500

$P(n) - \pi$ Para visualizar a diferença
-4.72645529925763696610374609008431434631347656250000
-1.75974790295185812283307313919067382812500000000000
0.00000000000000000000000000000000000000000000000000
0.00000000000000000000000000000000000000000000000000
0.00000000000000000000000000000000000000000000000000

Tabela 3. Convergência do π

O seguinte gráfico foi gerado com a análise dos result

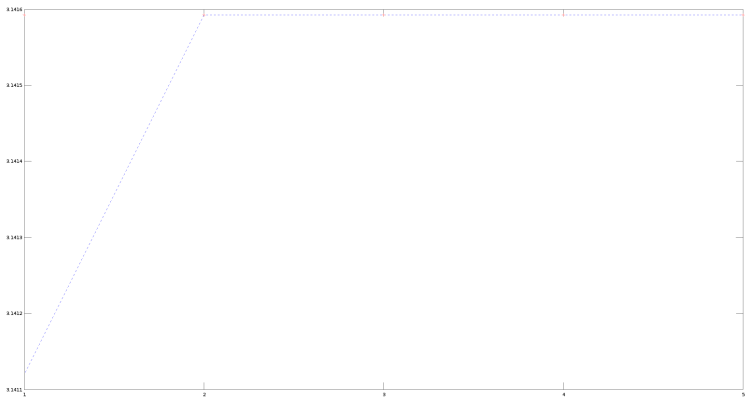

Figura 3. Convergência do π

O método iterativo utilizado foi descrito pelo seguinte código:

```
function [pif, pi_vec] = pi_it(iteration)
  pif(1) = 3 + sin(3);
  pi_vec(1) = pi
  for i = 2:iteration
    pif(i) = pif(i-1) + sin(pif(i-1));
    pi_vec(i) = pi;
    aux = pif(i);
  end
end
```

Referências

[Batista 2014] Batista, W. C. (2014). Métodos iterativos na resolução de equações.

[Cláudio 2000] Cláudio, D. M. (2000). *Cálculo Numérico Computacional (teoria e prática): algoritmos em pseudo-linguagem*.