Problemas resolvidos de MNUM - Teste 1

Diogo Miguel Ferreira Rodrigues
diogo.rodrigues@fe.up.pt

1 [2016T1] 2016 Teste 1

1.1 [2016T1-1] Problema 1

(a) O método da bisecção implica encontrar um intervalo inicial $[a, b]$ ao qual pertença a raiz da função, que, apesar de bastante simples, consegue ser mais complexo do que utilizar o método de Picard-Peano, que neste problema corresponde a encontrar a única recorrência possível

$$\frac{Bx-1}{x-1} = 0 \iff Bx - 1 = 0 \iff x = \frac{1}{B}$$

$$x_{n+1} = \frac{1}{B}$$

e aplicá-la uma única vez, dado que esta solução pelo método de Picard-Peano corresponde também à solução algébrica, para os valores de B em que a equação do enunciado é válida ($B \neq 0 \wedge B \neq 1$).
(b)

Ficheiro 1: Código-fonte 2016T1-1 (C++)

```cpp
#include <bits/stdc++.h>
using namespace std;
int main(){
    double B;
    cin >> B;
    cout << double(1.0L)/B << endl;
    return 0;
}
```

Ficheiro 2: Input 2016T1-1

```
0.629
```

Ficheiro 3: Output 2016T1-1

```
1.58983
```

1.2 Problema 3

(a) Espero encontrar duas soluções, uma próxima de $(-1.2, -0.6)$ e outra próxima de $(0.2, 1)$.
(b)

$$\begin{cases} f_1(x,y) = 0.7 + x - y \\ f_2(x,y) = 1 - x^2 - y \end{cases} \implies \mathbf{J} = \begin{bmatrix} f'_{1,x} & f'_{1,y} \\ f'_{2,x} & f'_{2,y} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -2x & -1 \end{bmatrix} \implies |\mathbf{J}| = -2x - 1$$

$$h = -\frac{\begin{vmatrix} f_1 & f'_{1,y} \\ f_2 & f'_{2,y} \end{vmatrix}}{|\mathbf{J}|} = -\frac{\begin{vmatrix} 0.7 + x - y & -1 \\ 1 - x^2 - y & -1 \end{vmatrix}}{|\mathbf{J}|} = -\frac{-x^2 - x + 0.3}{-2x - 1} = -\frac{x^2 + x - 0.3}{2x + 1}$$

$$k = -\frac{\begin{vmatrix} f'_{1,x} & f_1 \\ f'_{2,x} & f_2 \end{vmatrix}}{|\mathbf{J}|} = -\frac{\begin{vmatrix} 1 & 0.7 + x - y \\ -2x & 1 - x^2 - y \end{vmatrix}}{|\mathbf{J}|} = -\frac{x^2 + 1.4x - y - 2xy + 1}{-2x - 1} = \frac{x^2 + 1.4x - y - 2xy + 1}{2x + 1}$$

$$\begin{cases} x' = x + h \\ y' = y + k \end{cases} \iff \begin{cases} x' = x - \dfrac{x^2 + x - 0.3}{2x + 1} \\ y' = y + \dfrac{x^2 + 1.4x - y - 2xy + 1}{2x + 1} \end{cases} \iff \begin{cases} x' = \dfrac{x^2 + 0.3}{2x + 1} \\ y' = \dfrac{x^2 + 1.4x + 1}{2x + 1} \end{cases}$$

Para avaliar a condição de convergência, já colocámos as recorrências da equação anterior na forma necessária

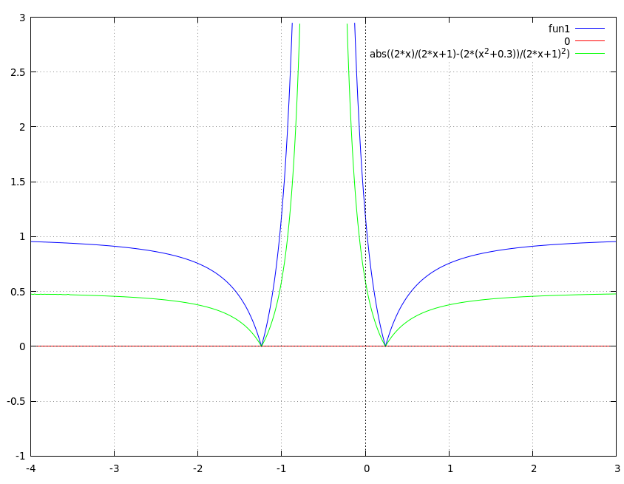
$$\begin{cases} x_{n+1} = g_1(x_n, y_n) \\ y_{n+1} = g_2(x_n, y_n) \end{cases}$$

$$g_{xx} = \left|\frac{\partial g_1}{\partial x}\right| + \left|\frac{\partial g_2}{\partial x}\right| = \left|\frac{2x}{2x+1} - \frac{2x^2 + 0.6}{(2x+1)^2}\right| + \left|\frac{2x^2 + 2.8x + 2}{(2x+1)^2} - \frac{2x + 1.4}{2x + 1}\right|$$

$$g_{yy} = \left|\frac{\partial g_1}{\partial y}\right| + \left|\frac{\partial g_2}{\partial y}\right| = |0| + |0| = 0$$

$$g_{xy} = \left|\frac{\partial g_1}{\partial x}\right| + \left|\frac{\partial g_2}{\partial y}\right| = \left|\frac{2x}{2x+1} - \frac{2x^2 + 0.6}{(2x+1)^2}\right| + |0| = \left|\frac{2x}{2x+1} - \frac{2x^2 + 0.6}{(2x+1)^2}\right|$$

Procedi à visualização de g_{xx}, g_{yy} e g_{xy} em gráfico com x como única variável independente, uma vez que y não influencia o valor de qualquer das funções.



Pode-se provar que, para valores $x < -1.1$ e $0.1 < x$, as funções possuem todas ordenadas inferiores a 1, mas o gráfico é suficientemente esclarecedor quanto a essa questão.
Assim, se $x < -1.1 \wedge y \in \mathbb{R}$ ou se $0.1 < x \wedge y \in \mathbb{R}$, pela condição de convergência o processo converge.
Escolhi o par ordenado $(x, y) = (-2, 0)$, por se encontrar dentro da primeira gama de valores de x para os quais o processo converge.
(c)

Ficheiro 4: Código-fonte 2016T1-3c (C++)

```cpp
#include <bits/stdc++.h>
using namespace std;
const double epsilon = 1e-8;
double fx(double x, double y){
    double up = x*x+0.3L;
    double dw = 2.0L*x+1.0L;
    return up/dw;
}
double fy(double x, double y){
    double up = x*x+1.4L*x+1.0L;
    double dw = 2.0L*x+1.0L;
    return up/dw;
}
int main(){
    double x, y; cin >> x >> y;
    double x_, y_;
    for(int i = 0; i < 1000; ++i){
        x_ = x; y_ = y;
        x = fx(x_, y_); y = fy(x_, y_);
        if(fabs(x-x_) < epsilon && fabs(y-y_) < epsilon) break;
    }
    cout << setprecision(7) << fixed << x << " " << y << endl;
    return 0;
}
```

Ficheiro 5: Input 2016T1-3c

```
-2.0 0.0
```

Ficheiro 6: Output 2016T1-3c

```
-1.2416198 -0.5416198
```

(d)

$$\begin{cases} x_{n+1} = -\sqrt{1-y_n} \\ y_{n+1} = 0.7 + x_n \end{cases} \qquad \begin{cases} x_{n+1} = -y_n - 0.7 \\ y_{n+1} = 1 - x_n^2 \end{cases}$$

Ficheiro 7: Código-fonte 2016T1-3di (C++)

```cpp
#include <bits/stdc++.h>
using namespace std;
double fx(double x, double y){
    return -sqrt(1.0L-y);
}
double fy(double x, double y){
    return 0.7L+x;
}
int main(){
    double x, y; cin >> x >> y;
    double x_, y_;
    int i;
    for(i=1; i<40; ++i){
        x_ = x; y_ = y;
        x = fx(x_, y_);
        y = fy(x_, y_);
        printf("%+.7f %+.7f "
               "%+.0E %+.0E\n",
               x, y, x-x_, y-y_);
    }
    printf("%d\n%+.32f\n%+.32f\n",
           i, x, y);
    for(; i < 10000000; ++i){
        x_ = x; y_ = y;
        x = fx(x_, y_);
        y = fy(x_, y_);
        if(x == x_ && y == y_)
            break;
    }
    printf("%d\n%+.32f\n%+.32f\n",
           i, x, y);
    return 0;
}
```

Ficheiro 8: Código-fonte 2016T1-3dii (C++)

```cpp
#include <bits/stdc++.h>
using namespace std;
double fx(double x, double y){
    return y-0.7L;
}
double fy(double x, double y){
    return 1-x*x;
}
int main(){
    double x, y; cin >> x >> y;
    double x_, y_;
    int i;
    for(i=1; i<40; ++i){
        x_ = x; y_ = y;
        x = fx(x_, y_);
        y = fy(x_, y_);
        printf("%+.7f %+.7f "
               "%+.0E %+.0E\n",
               x, y, x-x_, y-y_);
    }
    printf("%d\n%+.32f\n%+.32f\n",
           i, x, y);
    for(; i < 10000000; ++i){
        x_ = x; y_ = y;
        x = fx(x_, y_);
        y = fy(x_, y_);
        if(x == x_ && y == y_)
            break;
    }
    printf("%d\n%+.32f\n%+.32f\n",
           i, x, y);
    return 0;
}
```

Ficheiro 9: Input 2016T1-3d

```
0.0 0.5
```

Ficheiro 10: Output 2016T1-3di

```
-0.7071068 +0.7000000 -7E-01 +2E-01
-0.5477226 -0.0071068 +2E-01 -7E-01
-1.0035471 +0.1522774 -5E-01 +2E-01
-0.9207185 -0.3035471 +8E-02 -5E-01
-1.1417299 -0.2207185 -2E-01 +8E-02
-1.1048613 -0.4417299 +4E-02 -2E-01
-1.2007206 -0.4048613 -1E-01 +4E-02
-1.1852685 -0.5007206 +2E-02 -1E-01
-1.2250390 -0.4852685 -4E-02 +2E-02
-1.2187159 -0.5250390 +6E-03 -4E-02
-1.2349247 -0.5187159 -2E-02 +6E-03
-1.2323619 -0.5349247 +3E-03 -2E-02
-1.2389208 -0.5323619 -7E-03 +3E-03
-1.2378861 -0.5389208 +1E-03 -7E-03
-1.2405325 -0.5378861 -3E-03 +1E-03
-1.2401153 -0.5405325 +4E-04 -3E-03
-1.2411819 -0.5401153 -1E-03 +4E-04
-1.2410138 -0.5411819 +2E-04 -1E-03
-1.2414435 -0.5410138 -4E-04 +2E-04
-1.2413758 -0.5414435 +7E-05 -4E-04
-1.2415488 -0.5413758 -2E-04 +7E-05
-1.2415216 -0.5415488 +3E-05 -2E-04
-1.2415912 -0.5415216 -7E-05 +3E-05
-1.2415803 -0.5415912 +1E-05 -7E-05
-1.2416083 -0.5415803 -3E-05 +1E-05
-1.2416039 -0.5416083 +4E-06 -3E-05
-1.2416152 -0.5416039 -1E-05 +4E-06
-1.2416134 -0.5416152 +2E-06 -1E-05
-1.2416180 -0.5416134 -5E-06 +2E-06
-1.2416173 -0.5416180 +7E-07 -5E-06
-1.2416191 -0.5416173 -2E-06 +7E-07
-1.2416188 -0.5416191 +3E-07 -2E-06
-1.2416195 -0.5416188 -7E-07 +3E-07
-1.2416194 -0.5416195 +1E-07 -7E-07
-1.2416197 -0.5416194 -3E-07 +1E-07
-1.2416197 -0.5416197 +5E-08 -3E-07
-1.2416198 -0.5416197 -1E-07 +5E-08
-1.2416198 -0.5416198 +2E-08 -1E-07
-1.2416198 -0.5416198 -5E-08 +2E-08
40
-1.241619828926652679967901349300 52
-0.541619780732000011269633432675 62
83
-1.2416198487095662095214265718823 3
-0.5416198487095662539303475568885 9
```

Ficheiro 11: Output 2016T1-3dii

```
-0.2000000 +1.0000000 -2E-01 +5E-01
+0.3000000 +0.9600000 +5E-01 -4E-02
+0.2600000 +0.9100000 -4E-02 -5E-02
+0.2100000 +0.9324000 -5E-02 +2E-02
+0.2324000 +0.9559000 +2E-02 +2E-02
+0.2559000 +0.9459902 +2E-02 -1E-02
+0.2459902 +0.9345152 -1E-02 -1E-02
+0.2345152 +0.9394888 -1E-02 +5E-03
+0.2394888 +0.9450026 +5E-03 +6E-03
+0.2450026 +0.9426451 +6E-03 -2E-03
+0.2426451 +0.9399737 -2E-03 -3E-03
+0.2399737 +0.9411233 -3E-03 +1E-03
+0.2411233 +0.9424126 +1E-03 +1E-03
+0.2424126 +0.9418595 +1E-03 -6E-04
+0.2418595 +0.9412361 -6E-04 -6E-04
+0.2412361 +0.9415040 -6E-04 +3E-04
+0.2415040 +0.9418051 +3E-04 +3E-04
+0.2418051 +0.9416758 +3E-04 -1E-04
+0.2416758 +0.9415303 -1E-04 -1E-04
+0.2415303 +0.9415928 -1E-04 +6E-05
+0.2415928 +0.9416631 +6E-05 +7E-05
+0.2416631 +0.9416329 +7E-05 -3E-05
+0.2416329 +0.9415989 -3E-05 -3E-05
+0.2415989 +0.9416135 -3E-05 +1E-05
+0.2416135 +0.9416300 +1E-05 +2E-05
+0.2416300 +0.9416229 +2E-05 -7E-06
+0.2416229 +0.9416150 -7E-06 -8E-06
+0.2416150 +0.9416184 -8E-06 +3E-06
+0.2416184 +0.9416222 +3E-06 +4E-06
+0.2416222 +0.9416206 +4E-06 -2E-06
+0.2416206 +0.9416187 -2E-06 -2E-06
+0.2416187 +0.9416195 -2E-06 +8E-07
+0.2416195 +0.9416204 +8E-07 +9E-07
+0.2416204 +0.9416200 +9E-07 -4E-07
+0.2416200 +0.9416196 -4E-07 -4E-07
+0.2416196 +0.9416198 -4E-07 +2E-07
+0.2416198 +0.9416200 +2E-07 +2E-07
+0.2416200 +0.9416199 +2E-07 -9E-08
+0.2416199 +0.9416198 -9E-08 -1E-07
40
+0.2416198875900548714845683662133 5
+0.9416197865169004810326214283122 7
100
+0.2416198487095662650325778031401 6
+0.9416198487095662761348080493917 2
```

Como é evidente pelos resultados, ambos os conjuntos de expressões recorrentes convergiram para soluções, se bem que para soluções diferentes:

- (I) para $(-1.2416198487095662095214265718823 3, -0.5416198487095662539303475568885 9)$;
- (II) para $(+0.2416198487095662650325778031401 6, +0.9416198487095662761348080493917 2)$;

Apesar de ambos os métodos convergirem com velocidade comparável, inicialmente o método (II) parece convergir ligeiramente mais depressa até à iteração 26, quando parece que o método (I) ganha um pequeno avanço, sendo que a diferença entre iterações consecutivas se torna indiscernível para o método (I) à iteração 83, e para o método (II) à iteração 100.
De entre as operações utilizadas, $\sqrt{}$ é a que confere menor segurança em termos de implementação, sugerindo que o método (I) pode não ser tão confiável como o método (II) por utilizar $\sqrt{}$.
As soluções exatas em x são as soluções da equação

$$x^2 + x - 0.3 = 0 \iff x = \frac{-1 \pm \sqrt{1^2 - 4 * 1 * (-0.3}}{2 * 1} \iff x = \frac{-1 \pm \sqrt{2.2}}{2}$$

que, por Python3, dão os pares (x, y) com os valores:

- $(x_1, y_1) = (-1.2416198487095663248116281643122 7, -0.5416198487095663248116281643122 7)$
- $(x_2, y_2) = (0.2416198487095663248116281643122 7, 0.9416198487095663248116281643122 7)$

Os erros absolutos na última iteração são
em (I), $\varepsilon_x = 2.22E-16$ e $\varepsilon_y = 1.11E-16$ em (II), $\varepsilon_x = 5.55E-17$ e $\varepsilon_y = 4.87E-17$
que são compreensivelmente resultado do limite de precisão do double em C++ (até porque o nosso critério de paragem é a indescirnibilidade entre iterações consecutivas, que se verifica em computadores como consequência do limite da representação de números).
Os erros absolutos na 40ª iteração são
em (I), $\varepsilon_x = 1.98E-8$ e $\varepsilon_y = 6.80E-8$ em (II), $\varepsilon_x = 3.89E-8$ e $\varepsilon_y = 6.22E-8$
o que também suporta o facto de que (I) converge mais rapidamente que (II), uma vez que y depende apenas de x, x depende de si próprio e o erro absoluto à 40ª iteração em x é menor em (I) do que em (II).

2 [2017T1] 2017 Teste 1

2.1 [2017T1-1] Problema 1

$$f(x) = (x - 3.6) + \cos^3(x + 1.2)$$
$$f'(x) = 1 - 3\cos^2(x + 1.2)\sin(x + 1.2)$$

$$g(x) = x - \frac{f(x)}{f'(x)} = \frac{-\cos^2(x + 1.2)[3x\sin(x + 1.2) + \cos(x + 1.2)] + 3.6}{1 - 3\cos^2(x + 1.2)\sin(x + 1.2)}$$

Se definirmos

$$x' = x + 1.2$$
$$c = \cos(x')$$
$$s = \sin(x')$$
$$d = c^2$$

ficamos com

$$g(x) = \frac{-d(3xs + c) + 3.6}{1 - 3ds}$$

Ficheiro 12: Código-fonte 2017T1-1 (C++)

```cpp
#include <bits/stdc++.h>
using namespace std;
double g(double x){
    double x_ = x+1.2;
    double c = cos(x_);
    double s = sin(x_);
    double d = c*c;
    double up = -d*(3.0L*x*s+c)+3.6L;
    double dw = 1.0L-3.0L*d*s;
    return up/dw;
}
int main(){
    double x; cin >> x;
    cout << setprecision(10) << fixed << g(x) << endl;
    return 0;
}
```

Ficheiro 13: Input 2017T1-1

```
0.5
```

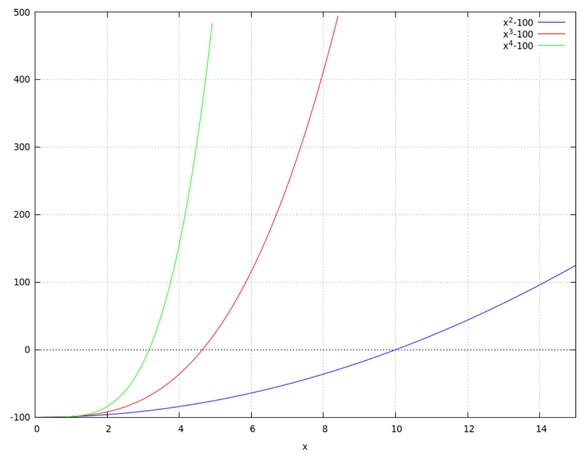
Ficheiro 14: Output 2017T1-1

```
3.7633057619
```

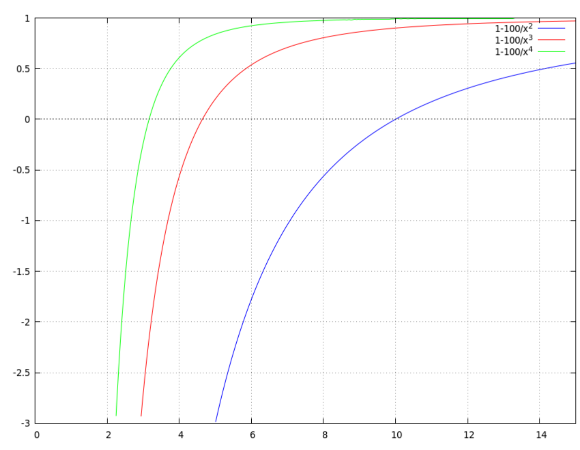
2.2 [2017T1-2] Problema 2

Utilizaria a fórmula **(a)**, uma vez que:

(a) Tem concavidade voltada para cima e é crescente em $x > 0$, pelo que se $x_n > \xi_x$ a tangente nesse ponto interseta o eixo das abcissas em $x_{n+1} > \xi_x$ mais próxima de ξ_x do que x_n. Agora, basta usar um guess $>= \xi_x$; se $x > 1$, o guess pode ser R, se $x \leq 1$ o guess pode ser 1.



(b) Tem concavidade voltada para baixo e é crescente em $x > 0$, pelo que teria que se verificar $0 < x_n < \xi_x$ para que a tangente nesse ponto intersete o eixo das abcissas em x_{n+1} que verifica $0 < x_{n+1} < \xi_x$ e que está mais próximo de ξ_x do que x_n; mas o guess também não pode ser demasiado próximo de 0, uma vez que o declive de $f(x) = 1 - R/x^m$ é muito elevado na proximidade de 0, o que significa que a convergência é lenta.



Em ambos os exemplos, são utilizados os parâmetros $R = 100$ e $m = 2, 3, 4$.
Em suma, é mais fácil determinar um guess para **(a)** do que para **(b)**, além de **(a)** convergir mais depressa do que **(b)**.

Ficheiro 15: Código-fonte 2017T1-2 (Python3)

```
def g(x, m, R):
    return x-x/m+R/(m*x**(m-1))
m = int(input())
R = float(input())
x = max(R,1.0)
i = 0
while i < 100:
    x_ = x
    x = g(x, m, R)
    print("%+.31f"%(x))
    if(x == x_): break
    i = i+1
```

Ficheiro 16: Input 2017T1-2

```
2
100.0
```

Ficheiro 17: Output 2016T1-2

```
+50.5000000000000000000000000000000
+26.2400990099009909783944749506190
+15.0255301199868132044912272249348
+10.8404346730269249832190325832926
+10.0325785109606044898100662976503
+10.0000528956426926185940828872845
+10.0000000001398969828869667253457
+10.0000000000000000000000000000000
+10.0000000000000000000000000000000
```

2.3 [2017T-5] Problema 5

Com truncagem.

op	±	mantissa			±	exp	comentários	
	+	1	3	3	+	0	$x = 2/15$	+0.133e+0
$*$	+	1	3	3	+	0	x	+0.133e+0
	+	1	7	6	-	1	$x^2 = 0.017689$	+0.176e−1
$*$	+	1	3	3	+	0	x	+0.133e+0
	+	2	3	4	-	2	$x^3 = 0.0023408$	+0.234e−2
	-	**2**	**0**	**0**	**+**	**1**	-2	−0.200e+1
	+	1	3	3	+	0	x	+0.133e+0
$*$	+	4	0	0	+	1	4	+0.400e+1
	+	**5**	**3**	**2**	**+**	**0**	$4x = 0.532$	+0.532e+0
	+	1	7	6	-	1	x^2	+0.176e−1
$*$	-	3	0	0	+	1	-3	−0.300e+1
	-	**5**	**2**	**8**	**-**	**1**	$-3x^2 = -0.0528$	−0.528e−1
	+	2	3	4	-	2	x^3	+0.234e−2
$*$	+	5	0	0	+	1	5	+0.500e+1
	+	**1**	**1**	**7**	**-**	**1**	$5x^3 = 0.0117$	+0.117e−1

$$
\begin{aligned}
x &= 0.133\\
x^2 &= 0.0176\\
x^3 &= 0.00234\\
5x^3 &= 0.0117\\
-3x^2 &= -0.0528\\
5x^3 - 3x^2 &= -0.0411\\
4x &= 0.532\\
5x^3 - 3x^2 + 4x &= 0.491\\
5x^3 - 3x^2 + 4x - 2 &= -1.51
\end{aligned}
$$

$$
\begin{aligned}
x &= 0.133\\
x^2 &= 0.0176\\
x^3 &= 0.00234\\
5x^3 &= 0.0117\\
-3x^2 &= -0.0528\\
4x &= 0.532\\
4x - 2 = -1.46 - 3x^2 + 4x - 2 &= -1.515x^3 - 3x^2 + 4x - 2 = -1.50
\end{aligned}
$$

diogo.rodrigues@fe.up.pt