TEA010 Matemática Aplicada I
Curso de Engenharia Ambiental
Departamento de Engenharia Ambiental, UFPR
Trabalho Computacional. Data de entrega: 24 de Abril de 2020.
Prof. Nelson Luís Dias

1 Instruções para a entrega do trabalho

Atenção: O grupo entregará apenas 2 arquivos digitais, que devem ser enviados para

`nldias@ufpr.br`,

com o assunto: `TEA010-tc-4-vanderp`. Os nomes dos arquivos serão

1. `tc-4-vanderp.py`,
2. `tc-4-vanderp.pdf` .

O seu programa deve estar escrito em Python $\geq 3.x$. O programa `tc-4-vanderp.py` deve gerar obrigatoriamente apenas um arquivo de saída, cujo nome será

`tc-4-vanderp.out`

O arquivo `tc-4-vanderp.pdf` deve conter suas respostas ao enunciado mais abaixo, figuras, equações utilizadas, referências bibliográficas, etc.. Não há regras muito fixas, mas você deve obedecer ao seguinte:

- 2,5 cm de margem (direita, esquerda, acima e abaixo),
- espaçamento simples,
- Tipo romano com serifa (por exemplo, Times Roman ou Times New Roman; **não use tipos sem serifa, tais como Arial ou Calibri**).

O arquivo `tc-4-vanderp.out` deve ser um arquivo texto; ele deve ser organizado em 3 colunas de 12 caracteres cada, impressas com o formato `%12.6f`, sem brancos adicionais entre as colunas. A primeira coluna deve conter os valores de μ, a partir de 0, em incrementos $\Delta\mu = 0{,}01$ até $\mu = 10$; a segunda coluna deve conter os valores de $T(\mu)$, e a terceira coluna os valores de $A(\mu)$ (vide seção 3), **nessa ordem**.

2 Correção do trabalho

Os critérios de correção são os seguintes

- Se o programa não rodar, por qualquer motivo, a nota do trabalho é zero.
- Se o programa não gerar os arquivos de saída com os nomes e o formato especificados, a nota é zero.
- Eu vou comparar graficamente o arquivo de saída contendo a sua solução com a solução numérica correta. As duas soluções devem concordar visualmente.
- O Português da apresentação deve ser correto. **Cuidado com erros de pontuação, concordância e ortografia**.
- A qualidade da apresentação deve ser boa. O texto deve ser claro, as referências a figuras e tabelas (se houver) devem seguir as regras acadêmicas usuais (figuras e tabelas numeradas, gráficos bem feitos, etc.).
- Pesquisa sobre o tema; referências adicionais; explorações adicionais do tema do trabalho; profundidade; etc., serão considerados na nota do trabalho.

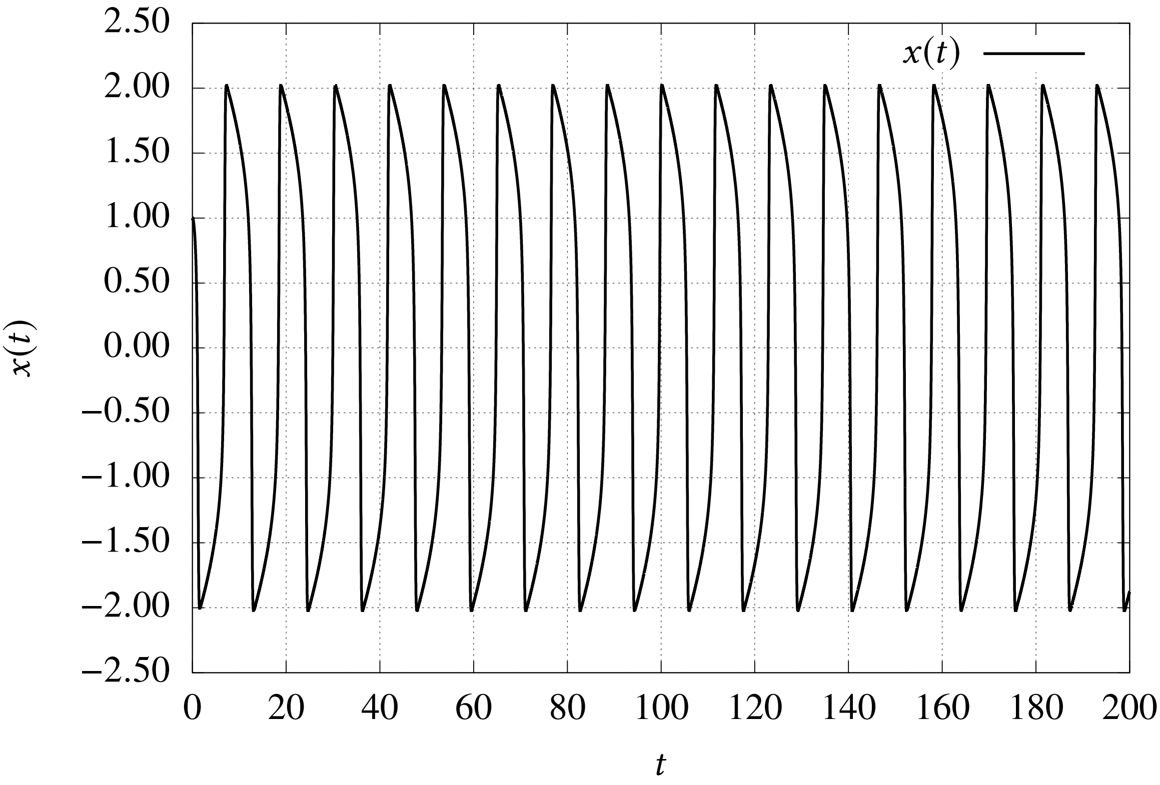

Figura 1: O oscilador não-linear de van der Pol ($\mu = 5$).

3 Trabalho computacional: O oscilador de van der Pol

A equação diferencial de van der Pol é

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} - \mu(1 - x^2)\frac{\mathrm{d}x}{\mathrm{d}t} + x = 0. \tag{1}$$

Ela pode ser reescrita como um sistema de duas equações diferenciais ordinárias:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = y, \tag{2}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \mu(1 - x^2)y - x. \tag{3}$$

Inicialmente, aprenda a resolver o sistema (2)–(3) numericamente com o método de Runge-Kuta de ordem 4. Use as condições iniciais $x(0) = 1$ e $y(0) = 0$; um passo de tempo $h = 0.01$, e $\mu = 5$. Plote o resultado até $t = 200$ (e portanto até $k = N = 20000$). Ele deve ser o mostrado na figura 1.

Agora, desenvolva uma método numérico para obter o período de oscilação $T(\mu)$ e a amplitude $A(\mu)$ de $x(t)$ em função de μ (note que, na figura 1, existe um transiente, até que o sistema entre em um regime estacionário). Atenção: não confunda estacionário com x constante! Você deve desprezar esse transiente antes de obter a amplitude e o período.

Isso pode ser feito da seguinte maneira:

1. Varie μ de 0 a 10 em incrementos de 0,01. Para cada μ, resolva (2)–(3) numericamente. O resultado serão os vetores $x = (x_0, x_1, \ldots, x_{N-1})$ e $y = (y_0, y_1, \ldots, y_{N-1})$ (em Python, esses serão os arrays `x` e `y`).

2. Obtenha $z = \mathrm{d}^2x/\mathrm{d}t^2$, a derivada segunda de x, numericamente. Você vai precisar dela para distinguir os mínimos dos máximos. Em Python, crie um array `z` com o mesmo tamanho de `x` e `y`, e faça

```
z[1:N-1] = (y[2:N] - y[0:N-2])/(2*h) # der 2a, numérica
z[0] = z[1]                          # sentinela
z[N-1] = z[N-2]                      # sentinela
```

3. Percorra a lista de x_k's para $k = 1, \ldots, N-1$: toda vez que $p = -x_{k-1}/x_k > 0$, a função $x(t)$ trocou de sinal. Sinalize isso com um novo i, e obtenha o zero da função $x(t)$ por interpolação linear:
$$\tau_i^x = t_{k-1} + \frac{p}{p+1}h.$$
Adicione τ_i^x a uma lista separada com os zeros de $x(t)$. Também vai ser útil guardar listas κ^{xe} e κ^{xd} com os índices k à esquerda e à direita dos zeros; para cada zero encontrado, seus i-ésimos elementos serão
$$\kappa_i^{xe} = k - 1,$$
$$\kappa_i^{xd} = k.$$
A partir de τ^x, calcule as diferenças entre os zeros consecutivos:
$$\delta_{i-1}^x = \tau_i^x - \tau_{i-1}^x, \ i \geq 1$$
Isso produz uma lista δ^x.

4. Repita o item 3 para o vetor y, obtendo também as listas τ^y, κ^{ye}, κ^{yd} e δ^y.

5. Descarte (por exemplo) os 4 primeiros valores de δ^x para evitar o transiente, e calcule a média sobre os i's restantes; a estimativa do período de $x(t)$ para o parâmetro μ será
$$T(\mu) = 2\overline{\delta_i^x}, \ i > 4.$$

6. Para as amplitudes, os extremos locais de x são os t's em que $y = 0$ (por quê?). No entanto, como acabamos de ver, devido à discretização nós temos apenas os pontos y_k antes (à esquerda) e depois (à direita) dos zeros. Como os índices k em que y cruza o eixo dos t's ficaram guardados nas variáveis κ^{ye} e κ^{yd}, uma estimativa razoável dos extremos de x (tanto os máximos locais quanto os mínimos locais) são as médias dos valores antes e depois desses zeros em y.

   Um algoritmo para encontrar os extremos em x portanto é o seguinte: Percorra as listas κ^{ye} e κ^{yd}. Para cada índice i dessas listas, calcule o valor médio de x em κ_i^{ye} e κ_i^{yd}. Adicione esse valor a uma lista `xex` com os valores extremos de x. Idem para z: cujos valores correspondentes devem ser armazenados em uma lista `zex`. Em Python (com numpy) isso pode ser feito muito eficientemente com

```
xex = (x[kyesq] + x[kydir])/2.0      # extremos
zex = (z[kyesq] + z[kydir])/2.0      # e suas curvaturas
```

   Finalmente, os máximos locais são os elementos `xex[i]` (para $i > 4$) em que `z[i]` < 0, e os mínimos locais são os elementos `xex[i]` em que `z[i]` > 0. A amplitude $A(\mu)$ é igual à metade da média dos máximos menos a média dos mínimos.

7. Prossiga para o próximo μ.

Os períodos obtidos em função de μ são mostrados na figura 2. As amplitudes em função de μ são mostradas na figura 3. Note que a amplitude aumenta rapidamente desde $A(0) = 1$ (que corresponde a uma equação diferencial linear) para $A(\mu) \gtrsim 2, \ \mu > 0.1$.

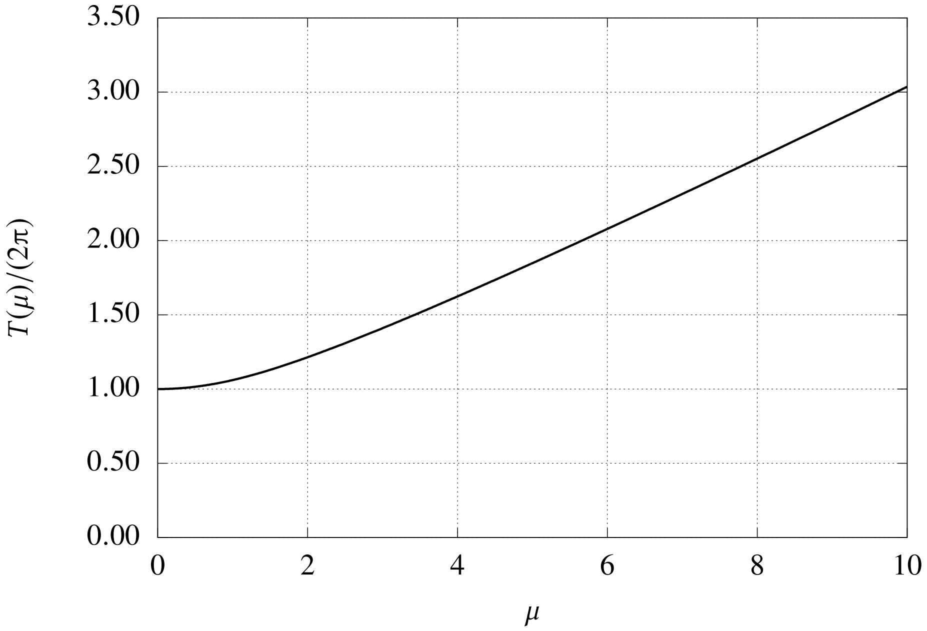

Figura 2: Período do oscilador de van der Pol em função de μ. .

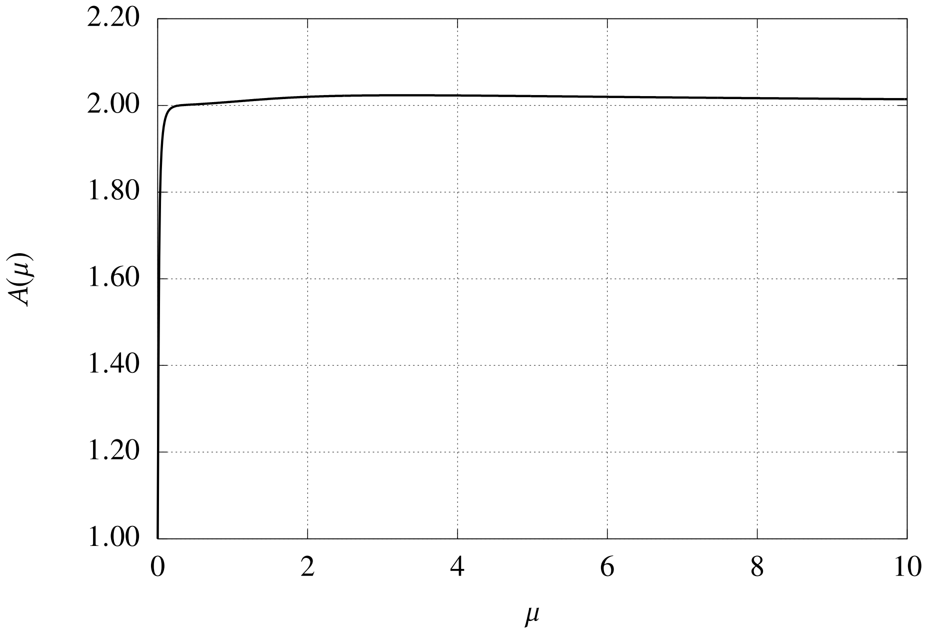

Figura 3: Amplitude do oscilador de van der Pol em função de μ. .