

# Modelação do consumo epidémico de drogas

## 1 Preâmbulo

A unidade curricular (UC) Laboratório/Projeto I tem por objetivo consolidar e estender os conhecimentos das unidades curriculares que funcionam no primeiro semestre da Licenciatura em Engenharia Informática do ISEP (LEI-ISEP). Esta UC inclui também um módulo de Competências pessoais e métodos de trabalho que visa preparar os estudantes para trabalhar eficazmente em equipas, melhorar a elaboração de relatórios e desenvolver competências de apresentação oral. Assim, com o projeto descrito neste enunciado pretende-se que os estudantes desenvolvam uma aplicação em linguagem *Java* (Horstmann, 2015), em grupo, aplicando conceitos da área da matemática. Pretende-se também que os estudantes elaborem um relatório que descreva a aplicação concebida, o processo de desenvolvimento e que apresentem e critiquem os resultados obtidos.

## 2 Plano de Trabalho

Segundo alguns estudos realizados, dependências como o alcoolismo e o consumo de drogas podem ser modelados como doenças infecciosas (Raza et al., 2022).

### 2.1 Modelação do Sistema de EDOs de 1ª ordem

Considerando que dependências podem ser modeladas como doenças infecciosas, poderemos utilizar o modelo SIR para a biologia populacional de doenças infecciosas, originalmente desenvolvido por Kermack e McKendrick em 1927 (Kermack & McKendrick, 1991), para modelar a dependência de drogas. No modelo SIR, a população está particionada em três grupos:

- Suscetíveis ao vício ( $s$ );
- Infetados pelo vício (incluindo reincidentes) que não estão em tratamento ( $i$ );
- Recuperados do vício ou os que estão em tratamento ( $r$ );

onde, num dado instante de tempo  $t \geq 0$ , temos  $s(t) \geq 0$ ,  $i(t) \geq 0$ ,  $r(t) \geq 0$  e  $s(t) + i(t) + r(t) \equiv N$ , sendo  $N$  o tamanho da população, que se supõe constante durante o período de estudo.

Para grandes populações, onde o tamanho exato da população pode não ser conhecido, usa-se, frequentemente, a formulação em termos de proporções da população, em vez dos valores absolutos. Assim, faz-se  $S(t) = s(t)/N$ ,  $I(t) = i(t)/N$  e  $R(t) = r(t)/N$ , com  $S(t) + I(t) + R(t) = 1$ . O modelo do consumo de drogas será, neste trabalho, o representado na Figura 1.

As variáveis e os parâmetros do modelo são então descritos como se segue:

Variáveis:

- $S(t)$ , proporção de indivíduos Suscetíveis ao vício, no instante de tempo  $t$ ;
- $I(t)$ , proporção de indivíduos Infetados pelo vício (inclui reincidentes), que não estão em tratamento, no instante de tempo  $t$ ;
- $R(t)$ , proporção de indivíduos Recuperados do vício e os que estão em tratamento, no instante de tempo  $t$ ;

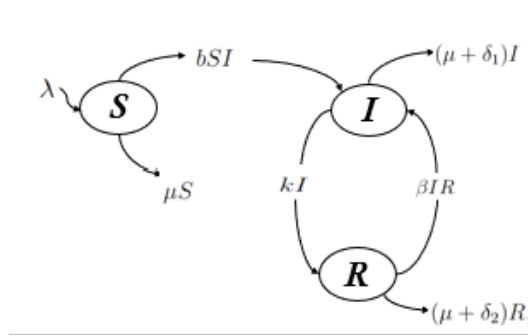


Figura 1: Representação do modelo.

com  $0 \leq S(t), I(t), R(t) \leq 1$  e  $S(t) + I(t) + R(t) = 1$ .

Parâmetros:

- $\lambda$ , a taxa de indivíduos suscetíveis a serem novos utilizadores de drogas;
- $b$ , a taxa de um indivíduo suscetível tornar-se utilizador de drogas;
- $k$ , a taxa de indivíduos que passam para a classe de recuperados, ou seja, que entram em tratamento ou recuperam do consumo de drogas;
- $\beta$ , a taxa de um indivíduo em tratamento ou recuperado ter uma recaída;
- $\mu$ , a taxa de mortalidade natural (não relacionada com o tratamento);
- $\delta_1$ , a taxa de remoção devido a mortes relacionadas com drogas em utilizadores que estão a consumir, ou a recuperação espontânea de indivíduos que pararam de consumir e não estão mais em estado suscetível;
- $\delta_2$ , a taxa de recuperação espontânea de utilizadores em tratamento ou que pararam de consumir e não são mais suscetíveis.

As equações diferenciais não lineares que representam a dinâmica acima descrita são as seguintes:

$$\frac{dS}{dt} = \lambda - bSI - \mu S, t \geq 0 \quad (1)$$

$$\frac{dI}{dt} = bSI - kI + \beta IR - (\mu + \delta_1)I, t \geq 0 \quad (2)$$

$$\frac{dR}{dt} = kI - \beta IR - (\mu + \delta_2)R, t \geq 0 \quad (3)$$

Para desenvolver este projeto o estudante deve, numa primeira fase, compreender os objetivos e estudar o modelo proposto de equações diferenciais ordinárias (EDOs) adotado para o problema em estudo.

Para a resolução do problema, dado não ter solução analítica, será necessário recorrer a métodos numéricos de resolução de equações diferenciais de 1<sup>a</sup> ordem: o método de Euler e o método de Runge-Kutta de 4<sup>a</sup> ordem. Após adquirir os conhecimentos necessários, o estudante terá de implementar uma aplicação que permita estudar e avaliar o consumo epidémico de drogas em determinadas condições. Esta aplicação deve incluir um conjunto de módulos que permitam: ler e escrever dados em ficheiros; implementar os métodos de Euler e Runge-Kutta de 4<sup>a</sup> ordem, implementar, visualizar e avaliar o modelo de propagação do vício (dado o estado inicial), utilizando os métodos acima referidos.

### 2.1.1 Condições iniciais do Sistema de EDOs de 1ª ordem

O sistema de EDOs tem um conjunto de condições iniciais que é necessário conhecer. A saber:

- As condições iniciais do modelo (SIR):  $S(0)$ ,  $I(0)$  e  $R(0)$ ;
- Os valores dos parâmetros do modelo;
- O número de dias para a previsão do modelo.

## 2.2 Implementação da solução do problema

Para a implementação do modelo, deverá ser utilizado um dos métodos de aproximação numérica, referidos na secção seguinte.

### 2.2.1 Aproximação numérica da solução do problema de valor inicial

Para resolver o sistema de EDOs, devem ser utilizados os métodos numéricos de Euler e de Runge-Kutta de 4ª ordem (Pina, 1995), que permitem obter uma solução aproximada para o problema de valor inicial (PVI) que se pretende estudar. Estes métodos consistem em substituir o processo de integração e derivadas de funções por operações aritméticas simples, conhecendo as condições iniciais do sistema.

### 2.2.2 Algoritmos de Euler e Runge-Kutta de 4ª ordem

O Algoritmo 1, abaixo, descreve o método de Euler para obter uma solução aproximada do PVI. O Algoritmo 2 descreve o método de Runge-Kutta para o mesmo efeito.

$$\frac{dy}{dx} = f(x, y),$$

com  $y(x_0) = y_0$ .

```

define function  $f(x, y)$ 
read values of initial condition ( $x_0$  and  $y_0$ )
read number of steps ( $n$ )
read size of step ( $h$ )
 $i \leftarrow 0$ 
do
     $y_n \leftarrow y_0 + h * f(x_0 + i * h, y_0)$ 
     $y_0 \leftarrow y_n$ 
     $i \leftarrow i + 1$ 
while  $i < n$ 
display  $y_n$  as result
  
```

**Algorithm 1:** Euler.

Ao usar métodos de aproximação numérica, visto não termos uma resolução com solução exata, é necessário considerar os erros de truncatura (ou discretização) que consistem no erro introduzido no passo de integração e os erros de arredondamento, que resultam dos cálculos efetuados. O erro global pode crescer de tal forma que se afaste qualitativamente da solução do problema com o evoluir dos cálculos ou provocar instabilidade (Fig. 2). A escolha do tamanho do passo  $h$  é importante.

```

define function  $f(x, y)$ 
read values of initial condition ( $x_0$  and  $y_0$ )
read number of steps ( $n$ )
read size of step ( $h$ )
 $i \leftarrow 0$ 
do
     $k_1 \leftarrow h * f(x_0, y_0)$ 
     $k_2 \leftarrow h * f(x_0 + h/2, y_0 + k_1/2)$ 
     $k_3 \leftarrow h * f(x_0 + h/2, y_0 + k_2/2)$ 
     $k_4 \leftarrow h * f(x_0 + h, y_0 + k_3)$ 
     $k \leftarrow (k_1 + 2 * k_2 + 2 * k_3 + k_4)/6$ 
     $y_n \leftarrow y_0 + k$ 
     $i \leftarrow i + 1$ 
     $x_0 \leftarrow x_0 + h$ 
     $y_0 \leftarrow y_n$ 
while  $i < n$ 
display  $y_n$  as result

```

**Algorithm 2:** Runge-Kutta de 4<sup>a</sup> ordem.

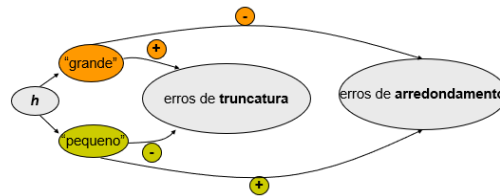


Figura 2: Representação dos erros numéricos.

### 2.3 Análise Comparativa

Deve ser realizada uma análise comparativa dos resultados obtidos pelos métodos numéricos para determinados valores de  $0 < h \leq 1$ . A aplicação deve incluir também um conjunto de funcionalidades que permitam realizar uma análise comparativa dos resultados obtidos para diferentes intervalos de tempo (número de dias). A aplicação deve disponibilizar funcionalidades para apoio às análises seguintes:

- Comparar os resultados obtidos e o comportamento do modelo, para um dado conjunto de parâmetros e condições iniciais, para diferentes intervalos de tempo, pelos métodos numéricos utilizados.

Os resultados devem ser guardados numa tabela, em ficheiro de texto, onde constem 5 colunas com os valores correspondentes a: dia, S, I, R e T (S+I+R).

## 3 Trabalho a Desenvolver

No trabalho a realizar devem ser consideradas as tarefas seguintes.

- Estudo/revisão de equações diferenciais de 1<sup>a</sup> ordem, lecionadas na UC AMATA. Constatação da existência de EDOs (e sistemas de EDOs) para as quais não se conhece um processo de

resolução analítica (exata). Compreender a necessidade de se recorrer a outros métodos de estudo, por exemplo os métodos numéricos para obtenção de soluções aproximadas.

- Estudo dos algoritmos do método de Euler e do método de Runge-Kutta de 4<sup>a</sup> ordem.
- Desenvolvimento de um módulo, `Euler()`, em Java, que implemente o algoritmo de Euler para resolução de EDOs.
- Desenvolvimento de um módulo, `RK4()`, em Java, que implemente o algoritmo de Runge-Kutta de 4<sup>a</sup> ordem para resolução de EDOs.
- Desenvolver uma aplicação informática que permita estudar a propagação de um vício numa dada população, para um número de dias e determinados parâmetros, usando um método numérico (Euler ou RK4) e um passo de integração numérica ( $h$ ).
- Desenvolver um módulo que permita gerar e gravar em ficheiro (formato txt/csv) e a representação gráfica (formato png) da variação das curvas S, I, R para uma dada população ao longo do tempo (número de dias) e para determinados parâmetros do modelo. Para a representação gráfica, é apenas permitido recorrer à aplicação gnuplot (Williams & Kelley, 2010), classe `java.lang.Runtime` e classes utilizadas na UC APROG. Não é permitida a utilização de bibliotecas que implementem interfaces para a biblioteca gnuplot, tais como as bibliotecas `jgnuplot` e `JavaPlot`.
- A aplicação deve permitir a execução de todas as funcionalidades através de um único comando.
- Elaborar um relatório onde consta:
  - Uma breve introdução às equações diferenciais de 1<sup>a</sup> ordem e sistemas SIR;
  - Uma descrição dos métodos numéricos de Euler e RK4 (apresentando os respetivos algoritmos);
  - Uma descrição da metodologia de trabalho utilizada para desenvolver a aplicação, referindo a abordagem *Scrum* e o planeamento do trabalho;
  - Uma descrição da implementação da aplicação, que deve incluir um diagrama que identifique claramente os módulos e suas dependências;
  - Uma análise dos resultados, onde devem ser apresentados e explorados os resultados obtidos. Deve ser analisada a evolução da propagação de um vício para um número de dias indicado. Devem apresentar e comentar os respetivos resultados apresentados em tabelas e graficamente (dando atenção à variação do tamanho do passo  $h$ ). Todos os resultados apresentados devem ser obtidos recorrendo apenas à aplicação desenvolvida pelo grupo.

### 3.1 Outros requisitos

- A aplicação deve ser desenvolvida em linguagem *Java* e estruturada e devidamente organizada em módulos. Será valorizada uma correta decomposição modular e o reaproveitamento de módulos. No final, o trabalho deve resultar um ÚNICO projeto.
- A aplicação deve ter uma interface simples e intuitiva que permita ao utilizador interagir com a aplicação de forma rápida e minimizando os erros. A interface da aplicação será em modo de texto e deverá incluir um conjunto de menus que permita aceder às funcionalidades de forma rápida, de acordo com o que é apresentado nos objetivos do projeto. Os parâmetros e os valores iniciais do modelo devem ser obtidos de ficheiros de texto (encontram um exemplo no Moodle).

- A aplicação pode ser executada em modo interativo ou sem interação (modo não interativo) por parte do utilizador.
  - No modo interativo, a aplicação deverá ser chamada da linha de comandos utilizando o seguinte comando: `java -jar nome_programa.jar`.
  - No modo não interativo, o utilizador especifica todos os parâmetros necessários à execução da aplicação na linha de comando: o ficheiro contendo os parâmetros necessários ao modelo SIR, outro ficheiro as condições iniciais do modelo, o método a usar e o passo de integração, o número de dias a considerar para análise e o nome do ficheiro de saída CSV onde será guardado o resultado obtido. Neste modo, o comando terá a seguinte sintaxe:  
`java -jar nome_programa.jar -b parametros.csv -c val_iniciais.csv -m X -p Y -d K`, em que:
    - \* o valor associado ao parâmetro  $m$  ( $X$ ) representa o método a usar (1-Euler ou 2-Runge Kutta de 4<sup>a</sup> ordem);
    - \* o valor associado ao parâmetro  $p$  ( $Y$ ) representa o passo de integração  $h$  (maior do que zero e menor ou igual a um);
    - \* o valor associado ao parâmetro  $d$  ( $K$ ) representa o número de dias a considerar para a análise (maior do que zero).
  - O nome do(s) ficheiro(s) de saída deve(m) permitir identificar a operação realizada.
  - Todos os métodos desenvolvidos devem estar associados a testes unitários. Por exemplo, se existir um método de leitura de parâmetros `ler_parametros(ficheiro)` também deve existir o método `test_ler_parametros(ficheiro, numParametrosEsperado)` (ver Algoritmo 1), em que `numParametrosEsperado` é o número de parâmetros lidos. Estes testes são extremamente úteis para determinar se os métodos estão de acordo com a sua especificação e se a edição destes não alterou a sua funcionalidade.

```
boolean test_ler_parametros(String ficheiro, int numParametrosEsperado){

    int numParametros = ler__parametros(ficheiro);

    if(numParametrosEsperado == numParametros)
        return True;

    return False;
}
```

Algoritmo 3: Exemplo de teste unitário.

## 4 Método de Trabalho

- Todas as equipas têm de utilizar a metodologia de trabalho definida no *eduScrum* (Delhij & Solingen, 2013) que tem sido adotada no desenvolvimento dos trabalhos do Módulo de Competências. Cada equipa tem de definir um *Scrum Master* e um suplente (para todo o projeto ou para cada semana). O Scrum Master será responsável pela gestão da execução das tarefas e representará a equipa nas reuniões com o cliente. Cada equipa terá um canal privado para o projeto, no *Teams*, a que todos os docentes terão acesso. É obrigatória a utilização da ferramenta *Trello* para o planeamento e registo de todas as tarefas, assim como a utilização

do canal privado no *Teams* para comunicação entre os membros da equipa e os docentes que orientam o projeto. A designação da board no *Trello* deve seguir o formato "LAPR1\_Turma\_Número-Equipa\_Projeto".

- A aplicação será desenvolvida utilizando o sistema de controle de versões *Git* e o *Bitbucket* (<https://bitbucket.org>). Todos os estudantes terão que criar uma conta no *Bitbucket* com o endereço de email do ISEP (i.e. XXXXXXX@isep.ipp.pt) e cada grupo terá que criar um repositório. A designação do repositório deve seguir o formato: "LAPR1\_DAB\_Grupo01". O repositório deve ser partilhado com todos os docentes que lecionam a turma onde o grupo está inserido.
- O grupo deve guardar todo o material desenvolvido para a realização do projeto na pasta Ficheiros do seu canal privado no Teams. Não é necessário incluir nesta pasta o código que está disponível no repositório do *BitBucket*.

## 5 Submissão do Trabalho

Datas e entrega de trabalho a efetuar através do Moodle:

- Dia 6 de janeiro de 2024, até às 23h00m:
  - Submeter o projeto desenvolvido, versão final, incluindo toda a estrutura de diretorias e ficheiros do projeto (incluindo o ficheiro jar a ser executado), num único ficheiro comprimido (ZIP).
  - Relatório em formato pdf não ultrapassando as 20 páginas. A escrita do relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de Competências).
  - O documento suporte da apresentação (que também deve estar no canal privado da equipa).

Nota: Os ficheiros deverão identificar, obrigatoriamente, a designação do grupo e a turma a que os alunos pertencem (Exemplo: "LAPR1\_DAB\_Grupo01\_projeto.ZIP" e "LAPR1\_DAB\_Grupo01\_relatorio.PDF").

## Referências

- Delhij, A., & Solingen, R. (2013). *The eduscrum guide: The rules of the game*. (Disponível em [http://eduscrum.nl/file/CKFiles/The\\_eduScrum\\_Guide\\_EN\\_December\\_2013\\_1.0.pdf](http://eduscrum.nl/file/CKFiles/The_eduScrum_Guide_EN_December_2013_1.0.pdf))
- Horstmann, C. (2015). *Big java: Early objects, 6th edition*. Wiley. Retrieved from <https://books.google.pt/books?id=ib12CwAAQBAJ>
- Kermack, W. O., & McKendrick, A. G. (1991). Contributions to the mathematical theory of epidemics–i. 1927. *Bulletin of mathematical biology*, 53(1–2), 33–55.
- Pina, H. (1995). *Métodos numéricos*. McGrawHill.
- Raza, A., Chu, Y.-M., Bajuri, M. Y., Ahmadian, A., Ahmed, N., Rafiq, M., & Salahshour, S. (2022). Dynamical and nonstandard computational analysis of heroin epidemic model. *Results in Physics*, 34(105245). Retrieved from <https://doi.org/10.1016/j.rinp.2022.105245>.
- Williams, T., & Kelley, C. (2010, March). *gnuplot 5.0: An interactive plotting program*. <http://gnuplot.sourceforge.net/>.