

Estudo da propagação de uma notícia falsa

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 visam 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

Para desenvolver este projeto o estudante deve, numa primeira fase, compreender os objetivos e estudar o modelo proposto de equações diferenciais ordinárias (EDOs) da propagação de uma notícia falsa. Para a resolução do problema, dado não ter solução analítica, será necessário recorrer a métodos numéricos de resolução de equações diferenciais de 1^a ordem: o método de Euler e o método de Runge-Kutta de 4^a ordem. Após adquirir os conhecimentos necessários, o estudante terá de implementar uma aplicação que permita estudar e avaliar a propagação de uma notícia falsa 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^a ordem, implementar, visualizar e avaliar o modelo de propagação de uma notícia falsa (dado o estado inicial), utilizando os métodos acima referidos.

2.1 Modelação da propagação de uma notícia falsa

O assunto das notícias falsas (fake news) tem-se tornado mais relevante com a utilização da tecnologia e das redes sociais, uma vez que aumenta a capacidade de indivíduos e grupos de comunicação social difundirem de notícias infundadas. A disseminação de desinformação na sociedade e as suas consequências tornou-se uma das questões mais discutidas e pertinentes da atualidade.

2.1.1 Definição do Sistema de EDOs de 1^a ordem

O modelo SIR para a biologia populacional de doenças infecciosas foi originalmente desenvolvido por Kermack e McKendrick em 1927 (Kermack, 1991). Este modelo descreve a interação (dinâmica) de três populações de indivíduos. A dos Suscetíveis (S): nesta fase o indivíduo está exposto à doença e pode ser infetado; dos Infetados (I): fase em que o indivíduo transmite a doença aos Suscetíveis mediante contactos; os Recuperados (R): fase em que o indivíduo deixa a fase de infeção (I).

O sistema de EDOs do modelo SIR apresenta a seguinte dinâmica populacional: um indivíduo da população S torna-se infetado com uma taxa de transmissão β ao interagir com a população I. É possível um indivíduo recuperar, passando a pertencer à população R, com uma taxa γ (Fig. 1).

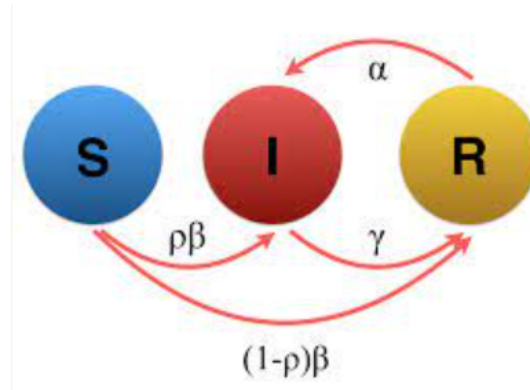


Figura 1: Representação do modelo SIR.

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

$$\begin{aligned}\frac{dS}{dt} &= -\beta.S.I \\ \frac{dI}{dt} &= \rho.\beta.S.I - \gamma.I + \alpha.R \\ \frac{dR}{dt} &= \gamma.I - \alpha.R + (1 - \rho).\beta.S.I\end{aligned}\tag{1}$$

Este modelo pode ser adaptado para descrever a propagação de uma notícia falsa. As populações S, I e R agora representam o seguinte: Suscetíveis(S)- nesta fase o indivíduo está exposto à notícia falsa e pode ser sugestionado; Infetados (I)- fase pela qual o indivíduo propaga notícia falsa aos Suscetíveis mediante contactos; Recuperados (R)- fase em que o indivíduo deixa a fase de acreditar na notícia falsa (I).

Os parâmetros do modelo agora são: β (taxa de propagação, ou contágio, da população), ρ (taxa da população não imune à notícia falsa quando a ouve), γ (taxa a que um Infetado rejeita a notícia falsa) e α (taxa de a população Recuperada ser re-Infetada).

2.1.2 Pressupostos do modelo

Neste modelo assume-se que a notícia propagada numa dada população N é falsa e que, como no modelo SIR, existem três populações, S , I e R , possíveis:

- Suscetíveis a uma notícia falsa (S) - ainda não ouviram a notícia falsa;
- Infetados pela notícia falsa (I) - ouviram a notícia falsa e acreditaram;
- Recuperadas da notícia falsa (R) - ouviram a notícia falsa e não acreditam.
- A notícia falsa propaga-se via interação direta com o infetado e há uma quantidade **fixa** de pessoas no sistema, $N = S + I + R$.

2.1.3 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:

- O tamanho, N , da população (eg. o número de estudantes do DEI);

- A propagação da notícia falsa é iniciada por **1** pessoa (considera-se $S = N - 1$, $I = 1$, $R = 0$);
- O número de dias para a previsão do modelo;
- Os parâmetros do modelo: α , β , γ e ρ .

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 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 passo h é importante. Note-se ainda que o modelo que estamos a estudar consiste num sistema de três EDOs e não apenas uma equação.

```

define function  $f(x, y)$ 
read values of initial condition ( $x_0$  and  $y_0$ )
read number of steps ( $n$ )
read 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 4a ordem

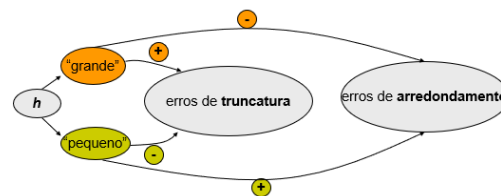


Figura 2: Representação dos erros numéricos.

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, para um dado conjunto de parâmetros, tamanho da população N e número de dias, pelos métodos numéricos utilizados;
- Comparar o comportamento do modelo SIR quando considerados diferentes conjuntos de parâmetros do modelo (α , β , γ e ρ).

Os resultados devem ser guardados numa tabela, em ficheiro de texto, onde constem 4 colunas com os valores correspondentes a: dia, S, I, R e N.

3 Trabalho a Desenvolver

No trabalho a realizar devem ser consideradas as tarefas seguintes.

- Estudo/revisão de equações diferenciais de 1^a 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). 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^a 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^a ordem para resolução de EDOs.
- Desenvolver uma aplicação informática que permita estudar a propagação de uma notícia falsa numa população (na LEI-ISEP) de um dado tamanho ($\simeq 1000$), 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, a representação gráfica (formato png) das curvas S, I, R para uma dada população ao longo do tempo (número de dias) e determinados parâmetros do modelo. Para implementar este módulo é permitido apenas 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^a 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;
 - 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 para dois casos de estudo: o "Zé Seguro" e "Maria Bocas". Deve ser analisada a evolução da propagação da notícia falsa no período de dias e tamanho da população pedidos. Devem apresentar e comentar os respetivos resultados apresentados em tabelas e graficamente (dando atenção à variação do passo h). Serão disponibilizados no Moodle os ficheiros contendo os dados necessários ao estudo. Todos os resultados apresentados devem ser obtidos recorrendo à aplicação desenvolvida pelo grupo.

3.1 Outros requisitos

- A aplicação deve ser desenvolvida em linguagem *Java* e estruturada e organizada em módulos. Será valorizada uma correta decomposição modular e o reaproveitamento de módulos. No final do 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. A interface da aplicação também deve permitir parametrizar cada uma das funcionalidades selecionadas, de acordo com o que é apresentado nos objetivos do projeto. Os parâmetros relativos ao modelo devem ser lidos de um ficheiro 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, o método a usar e o passo de integração, o tamanho da população em estudo, o número de dias a considerar para análise e o nome do ficheiro de saída CSV onde é guardado o resultado obtido. Neste modo, o comando terá a seguinte sintaxe: `java -jar nome_programa.jar ficheiroSIR.csv -m X -p Y -t Z -d K`, em que:
 - * o valor associado ao parâmetro m (X) representa o método a usar (1-Euler ou 2-Runge Kutta de 4^a ordem);
 - * o valor associado ao parâmetro p (Y) representa o passo de integração h (maior do que zero e menor do que um);
 - * o valor associado ao parâmetro t (Z) representa o tamanho da população em estudo N (maior do que zero);
 - * 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. Por exemplo, um nome possível de um dos ficheiros de texto de saída para o comando `java -jar nome_programa.jar ficheiroSIR.csv -m 1 -p 0.1 -t 1000 -d 30`, para o exemplo que consta no Moodle, seriam produzidos os seguintes ficheiros `Dinam1p01t1000d30.csv` e `Ru-cam1p01t1000d30.csv`

4 Método de Trabalho

- Todos os estudantes devem utilizar a metodologia de trabalho definida no *eduScrum* (Delhij & Solingen, 2013). Cada um dos grupos deve escolher um *Scrum Master* (para todo o projeto ou para cada semana do projeto), que será responsável por gerir a execução de tarefas. Este assunto deve ser combinado com as docentes do módulo de Competências. Para atingir os objetivos, o grupo deve utilizar a ferramenta *Trello* e registar as tarefas do projeto, a atribuição de tarefas, o estado de cada tarefa e as tarefas concluídas.
- 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 8 de janeiro de 2023, 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 25 páginas. A escrita do relatório deve seguir as instruções formais e o modelo disponibilizado nas aulas TP (módulo de Competências).

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)
- Horstmann, C. (2015). *Big java: Early objects, 6th edition*. Wiley. Retrieved from <https://books.google.pt/books?id=ib12CwAAQBAJ>
- Kermack, . M. A. G., W. O. (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.
- Williams, T., & Kelley, C. (2010, March). *gnuplot 5.0: An interactive plotting program*. <http://gnuplot.sourceforge.net/>.