

# LEIM – Modelação e Simulação de Sistemas Naturais

Época Especial

19 fevereiro 2019

## Grupo I – Sistemas Dinâmicos (5.0 val.)

Considere a seguinte equação que representa a evolução de uma determinada população:

$$p_{n+1} = p_n + ap_n - \frac{a}{M} p_n^2$$

- Represente este sistema através de um diagrama de Níveis e Fluxos. Quantas variáveis de estado tem o sistema? E quantas variáveis de fluxo? E variáveis auxiliares? [1.0]
- Qual o significado das duas constantes  $a$  e  $M$ ? Considere uma população inicial,  $p_0 = 200$ ,  $a = 3$ ,  $M = 300$ . Determine o valor da população para  $n = 2$ . Comente. [1.0]
- Trace um gráfico que mostre a relação entre o valor das populações em duas iterações consecutivas,  $p_n$  e  $p_{n+1}$ . Trata-se de um sistema linear ou não linear? Justifique. [1.0]
- Quantos pontos fixos tem o sistema? Determine-os e classifique-os quanto à estabilidade. Justifique. [1.0]
- Este sistema poderá exibir, nalgumas circunstâncias, comportamento caótico? Como identificaria esse comportamento? [1.0]

## Grupo II – Autómatos Celulares (5.0 val.)

Considere um autómato celular 1D, topologia toroidal, binário, com um sistema de vizinhança assimétrica, conforme explicitado na figura

	o	o	o	x	o		
--	---	---	---	---	---	--	--

- Determine a dimensão da tabela de regra para cada uma das três situações: i) caso geral; ii) regra totalística; iii) regra totalística externa [1.5]
- No caso da regra ser totalística externa, quantas seriam as regras possíveis de definir? [0.5]
- Considere a aplicação da regra da maioria (incluindo a própria célula). Esta regra pode ser considerada totalística? Justifique. [1.0]
- Escreva a respetiva tabela de regra (relativa à alínea anterior) [1.0]
- Usando a referida regra, determine a evolução de um autómato com 6 células, a partir da seguinte configuração inicial. Converge para ponto fixo ou ciclo limite? Justifique. [1.0]

1	1	1	0	0	0
---	---	---	---	---	---

## Grupo III – Agentes Autónomos (5.0 val.)

Considere os quatro seguintes agentes autónomos que evoluem de acordo com um comportamento de grupo, usualmente designado por *flocking*, sendo que todos os agentes têm massa igual a 2 e visão omnidirecional, com raio igual a 10.

	Atributos (t=0)
Agente 1	Localização: (10, 4), Velocidade: (-0.4, 0.8)
Agente 2	Localização: (0, 0), Velocidade: (0.5, 0.5)
Agente 3	Localização: (13, 0), Velocidade: (0, 1)
Agente 4	Localização: (8, -1), Velocidade: (-0.5, 0.5)

- Represente graficamente, no espaço 2D, os quatro agentes (posições e velocidades). [1.0]

- b) No algoritmo *flocking* são combinados 3 comportamentos individuais: *separate*, *align* e *cohesion*. Descreva, sucintamente, cada um destes comportamentos e explique a forma como estes são combinados [1.0]
- c) Considere o comportamento *cohesion*. Qual a velocidade desejada do agente 1, no que se refere a esse comportamento? E qual respetiva força de guiamento? Justifique. [1.0]
- d) Relativamente à alínea anterior e supondo um ritmo de simulação de 2 fps, determine a nova localização e velocidade, para o agente 1. Justifique os cálculos realizados. [1.0]
- e) Considere agora que em vez de *flocking* o agente 1 está a perseguir o agente 2 usando o comportamento *pursuit* (com  $\Delta T = 2s$ ). Determine a respetiva força de guiamento, sabendo que o módulo da velocidade desejada está limitado a um valor de 5. Justifique. [1.0]

#### Grupo IV – Programação (5.0 val.)

Considere o seguinte troço de código, relativo à implementação de um ecossistema.

1	public class Animal {
2	protected PVector loc, vel;
3	public Animal(PVector loc) {
4	this.loc = loc;
5	vel = PVector(0,1);}
6	public void move() {
7	loc.add(vel);}}
8	
9	public class Prey extends Animal {
10	private float energy;
11	public Prey(PVector position) {
12	super(position);
13	energy = 15.; }
14	public void move() {
15	super.move();
16	energy -= 2; }}

- a) O código acima não compila. Identifique o erro e corrija-o. [1.0]
- b) Quantos atributos tem um objeto da classe **Prey**? Enumere-os. Justifique. [1.0]
- c) Considere o seguinte código. Qual o resultado da execução do método **println** na última linha? Justifique. [1.0]

```
PVector position = new PVector(10,40);
Prey p1 = new Prey(position);
Prey p2 = new Prey(position);
p1.move();
println(p1.loc, p2.loc);
```

- d) Altere o código do método **move** (da classe **Animal** ou **Prey**) de modo que uma presa fique com a sua posição inalterada quando a sua energia é igual ou inferior a 5. [1.0]
- e) Escreva o código do método **eat** supondo que as presas apenas encontram alimento, com valor nutritivo igual a 2, quando estão na zona sul do mundo (na metade inferior do ecrã). [1.0]