Projeto de sistemas dinâmicos homogêneos lineares de primeira ordem com coeficientes constantes de duas dimensões

CM116 - Projeto 2

Entrega: 8 de Maio

1 Introdução

Vamos considerar o sistema

$$\begin{cases} x'(t) &= a_{11}x(t) + a_{12}y(t) \\ y'(t) &= a_{21}x(t) + a_{22}y(t) \end{cases}$$

Que também pode ser escrito como

$$Y'(t) = AY(t),$$

onde $Y(t) = (x(t), y(t))^T$ e A é uma matriz. Por acaso, a solução desse sistema depende dos autovalores e autovetores.

- Os autovalores são reais e diferentes;
- Os autovalores são reais e iguais com dois autovetores linearmente independentes associados;
- Os autovalores não são reais (são complexos);
- Os autovalores são reais e iguais com um único autovetor associado;

Além disso, o sinal do autovalores (ou da parte real deles) também influencia o resultado.

2 Trabalho

Este trabalho é individual. A entrega será no segunda **09 de Maio, até 12h00**. Cada aluno deve fazer alguns exemplos seguindo a distribuição abaixo. Façam imagens e animações, e usem vários pontos iniciais. Usem as funções adicionais da próxima seção.

- **Francine**
  - Autovalores reais diferentes positivos;
  - Autovalores complexos com parte real nula;
- **Oksana**
  - Autovalores iguais negativos com dois autovetores;
  - Autovalores complexos com parte real positiva;
- **Jaqueline**
  - Autovalores iguais positivos com um autovetor;
  - Autovalores iguais negativos com um autovetor;

- **Daniel**
  - Autovalores reais diferentes negativos;
  - Autovalores iguais positivos com dois autovetores;
- **Adrean**
  - Autovalores reais diferentes um positivo e um negativo;
  - Autovalores complexos com parte real negativa;

3 Parte computacional

Todos as implementações devem ser feitas considerando $t_0 = 0$.

Todos as soluções numéricas devem ser feitas com um método Runge-Kutta de 4 ordem:

```
function rk4(A, Y0, tf; N = 10000)
  t = linspace(0, tf, N)
  h = t[2]
  n = length(Y0)
  Y = zeros(n, N)
  Y[:,1] = Y0
  for i = 1:N-1
    y = Y[:,i]
    k1 = A*y
    k2 = A*(y + h*k1/2)
    k3 = A*(y + h*k2/2)
    k4 = A*(y + h*k3)
    Y[:,i+1] = y + h*(k1+k2+k3+k4)/6
  end
  return t, Y
end
```

Para fazer uns gráficos interessantes, faça um plot do campo vetorial também. Use o código abaixo para isso:

```
function quiver_plot(A, xmin, xmax, ymin, ymax; N = 20)
  x = linspace(xmin, xmax, N + 2)
  y = linspace(ymin, ymax, N + 2)
  h = [x[2]-x[1]; y[2]-y[1]]
  V = Array{Array{Float64}, 2}(N,N)
  for i = 1:N
    for j = 1:N
      V[i,j] = A*[x[i+1]; y[j+1]]
    end
  end
  M = maximum([norm(v) for v in V])
  plot()
  for i = 1:N
    for j = 1:N
      v = h.*V[i,j]/M
      xs = [x[i+1]; x[i+1]+v[1]]
      ys = [y[j+1]; y[j+1]+v[2]]
      plot!(xs, ys, leg=false, c=:gray)
```

```
      scatter!(xs[2:2], ys[2:2], c=:gray, m=:xcross, ms=3)
    end
  end
end
```

Para fazer a gravação de um gif, você pode fazer algo do tipo

```
using Plots
gr()

anim = Animation()
for i = 1:10
  # Gera 100 pontos na tela. Modifique pelos seus plots
  scatter(rand(100), rand(100), leg=false)
  frame(anim)
end
gif(anim, "exemplo.gif", fps=20)
```

Boa Sorte