**ListaParaProva1\_tri2\_2025x.pdf**

**Semana 23 - 25/06/25**

<https://drive.google.com/file/d/1wONJa6OZS4OJkodhlEbe118AjmJj7_kb/view>

**menu.m**

% ListaParaProva1\_tri2\_2025x.pdf

% Semana 23 - 25/06/25

% Lucas Vasconcelos Resende - 205B

clear all; close all; clc;

opção = 1;

while (opcao != 0)

printf("1 até 4 = questao 1,... (Geometria, Figuras, Aplicacoes em Fisica)\n");

printf("5 até 8 = questao 2,... (Polinomios)\n");

printf("9 = questão 3.1 (Strings)");

printf("10 até 11 = qustao 4,... (Vetores)");

printf("12 até 15 = questao 5,... (Analise de Dados e Ajustes de Curvas)");

printf("16 até 19 = questao 6,... (Introducao a Derivadas)");

printf("20 até 22 = questao 7,... (Sinais, Amostragem, Aliasing)");

opcao = input('digite o exercício ou 0 para sair: ');

close all

switch(opcao)

case 1

exercício1

case 2

exercicio2

case 3

exercicio3

case 4

exercicio4

case 5

exercicio5

case 6

exercício6

case 7

exercicio7

case 8

exercicio8

case 9

exercicio9

case 10

exercicio10

case 11

exercício11

case 12

exercicio12

case 13

exercicio13

case 14

exercicio14

case 15

exercicio15

case 16

exercício16

case 17

exercicio17

case 18

exercicio18

case 19

exercicio19

case 20

exercicio20

case 21

exercicio21

case 22

exercicio22

endswitch

endwhile

**exercicio1.1.m**

%Lista para a prova 2tri - Mateus Leon 205

%triongulo

hipotenusa = 20;

theta = input('Digite o ângulo de inclinação: ');

tr = theta \* pi / 180; %conversão para radiano, para funções trigonométricas

A = [0; 0];

B = [hipotenusa\*cos(tr); 0];

C = [hipotenusa\*cos(tr); hipotenusa\*sin(tr)];

x = [A(1), B(1), C(1), A(1)]; %os xizes

y = [A(2), B(2), C(2), A(2)]; %os ipsulons

%caixa em cima

caixalen = 2;

Ac = [(hipotenusa/2-caixalen/2)\*cos(tr); (hipotenusa/2-caixalen/2)\*sin(tr)];

Bc = [(hipotenusa/2+caixalen/2)\*cos(tr); (hipotenusa/2+caixalen/2)\*sin(tr)];

linhaperp = caixalen \* [cos(tr+pi/2); sin(tr+pi/2)];

Dc = Ac + linhaperp;

Cc = Bc + linhaperp;

xc = [Ac(1), Bc(1), Cc(1), Dc(1), Ac(1)];

yc = [Ac(2), Bc(2), Cc(2), Dc(2), Ac(2)];

meio = (Ac+Bc+Cc+Dc)/4;

%vetores

vtam = 3;

%perpendicular da caixa

vperp = vtam \* [cos(tr+pi/2); sin(tr+pi/2)];

Ap = (Ac+Bc)/2 + vperp;

Bp = (Ac+Bc)/2 - vperp;

xp = [Ap(1), Bp(1)];

yp = [Ap(2), Bp(2)];

%gravidade

Ag = [hipotenusa/2\*cos(tr); hipotenusa/2\*sin(tr)+vtam];

Bg = [hipotenusa/2\*cos(tr); hipotenusa/2\*sin(tr)-vtam];

xg = [Ag(1), Bg(1)];

yg = [Ag(2), Bg(2)];

%angulo1

raio1 = 2;

ang1 = linspace(0, tr, 50);

ang1x = raio1 \* cos(ang1) + A(1);

ang1y = raio1 \* sin(ang1) + A(2);

%angulo2

raio2 = 1;

ang2 = linspace(-pi/2, tr-pi/2, 50);

ang2x = raio2 \* cos(ang2) + (Ag(1)+Bg(1))/2;

ang2y = raio2 \* sin(ang2) + (Ag(2)+Bg(2))/2;

plot(x, y, '-ob', 'MarkerSize', 7);

hold on

plot(xc, yc, '-r', 'MarkerSize', 3);

hold on

plot(meio(1), meio(2), '.r', 'MarkerSize', 10);

hold on

plot(xp, yp, '--k', 'LineWidth', 1.2);

hold on

plot(xg, yg, '--k', 'LineWidth', 1.2);

hold on

plot(ang1x, ang1y, 'm', 'LineWidth', 1.5);

dist1 = raio1/8;

text(ang1x(length(ang1x)/2)+dist1, ang1y(length(ang1y)/2)+dist1, '\theta', 'FontSize', 10, 'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'left');

hold on

plot(ang2x, ang2y, 'm', 'LineWidth', 1.5);

dist2 = 0.8;

text((Ag(1)+Bg(1))/2+dist2, (Ag(2)+Bg(2))/2-dist2\*1.5, '\theta', 'FontSize', 10, 'VerticalAlignment', 'top', 'HorizontalAlignment', 'left');

grid minor on

axis equal

**exercicio1.2.m**

%Questoes para a prova - Mateus Leon 205

%Lista 1

opt = 'a';

while opt != 'c'

printf("Escolha uma das opções a ser plotada -\n\

(a) poligono inscrito\n\

(b) poligono circunscrito\n\

(c) sair\n");

printf("Digite uma das opções: ");

opt = scanf('%s', opt);

switch opt

case 'a'

printf("\n=======================\n\

polígono inscrito\n\

=======================\n");

n = input('Digite o número de lados, maior que 2: ');

while n <= 2

n = input('Digite outro número de lados, maior que 2: ');

endwhile

raio = input("Digite o raio da circunferência: ");

centro(1) = input('Digite a coordenada X do centro: ');

centro(2) = input('Digite a coordenada Y do centro: ');

theta = linspace(0, 2\*pi, 100);

xc = centro(1) + raio\*cos(theta);

yc = centro(2) + raio\*sin(theta);

figure

plot(xc, yc, 'r-');

hold on

poligonoang = linspace(0, 2\*pi, n+1);

xp = centro(1) + raio \* cos(poligonoang);

yp = centro(2) + raio \* sin(poligonoang);

xp(length(xp)+1) = xp(1);

yp(length(yp)+1) = yp(1);

plot(xp, yp, '-o');

axis equal

grid minor on

case 'b'

printf("\n=======================\n\

polígono circunscrito\n\

=======================\n");

n = input('Digite o número de lados, maior que 2: ');

while n <= 2

n = input('Digite outro número de lados, maior que 2: ');

endwhile

raio = input("Digite o raio da circunferência: ");

centro(1) = input('Digite a coordenada X do centro: ');

centro(2) = input('Digite a coordenada Y do centro: ');

theta = linspace(0, 2\*pi, 100);

xc = centro(1) + raio\*cos(theta);

yc = centro(2) + raio\*sin(theta);

figure

plot(xc, yc, 'r-');

hold on

raiopoligono = raio/cos(pi/n);

poligonoang = linspace(0, 2\*pi, n+1);

xp = centro(1) + raiopoligono \* cos(poligonoang);

yp = centro(2) + raiopoligono \* sin(poligonoang);

xp(length(xp)+1) = xp(1);

yp(length(yp)+1) = yp(1);

plot(xp, yp, '-o');

axis equal

grid minor on

case 'c'

printf("Saindo...\n")

otherwise

printf("Opção inválida.\n");

endswitch

endwhile

**exercicio1.3.m**

%Questoes para a prova - Mateus Leon 205

%Lista 1

printf("Digite o comprimento de três lados para um triângulo\n")

lado1 = input("Digite o lado 1: ");

lado2 = input("Digite o lado 2: ");

lado3 = input("Digite o lado 3: ");

if (lado1 + lado2 > lado3) && (lado1 + lado3 > lado2) && (lado2 + lado3 > lado1)

disp('Triângulo válido.');

A = [0; 0];

B = [lado3; 0];

xc = (lado3^2 + lado2^2 - lado1^2) / (2 \* lado3);

yc = sqrt(lado2^2 - xc^2);

C = [xc; yc];

x = [A(1), B(1), C(1), A(1)];

y = [A(2), B(2), C(2), A(2)];

%triongulo

plot(x, y, '-g');

hold on

%barissentro

sentro = (A+B+C)/3;

plot(sentro(1), sentro(2), 'og')

hold on

%medianum

m1p = (A+B)/2;

m1x = [m1p(1); xc];

m1y = [m1p(2); yc];

plot(m1x, m1y, 'r')

hold on

%medianois

m2p = (B+C)/2;

m2x = [m2p(1); 0];

m2y = [m2p(2); 0];

plot(m2x, m2y, 'r')

hold on

%medianeis

m3p = (C+A)/2;

m3x = [m3p(1); lado3];

m3y = [m3p(2); 0];

plot(m3x, m3y, 'r')

hold off

grid on

else

disp('Não é possível formar um triângulo com esses lados.');

end

**exercicio1.4.m**

% ListaParaProva1\_tri2\_2025x.pdf

% Semana 23 - 25/06/25

% Lucas Vasconcelos Resende - 205B

clear all; clc; close all;

n = randi([1,12]); %numero aleatórios de circulos (entre 1 e 12) / gera só um numero

raio = randi([1,10]); %raio dos cículos / gera só um numero

x = randi([1,40], 1, n); %coordenada x dos círculo / gera n numeros

y = randi([1,40], 1, n); %coordenada y dos círculo / gera n numeros

figure(1)

hold on

axis equal;

for i = 1 : n

%cálculos pro circulo

theta = linspace(0, 2\*pi, 100);

xc = x(i) + raio \* cos(theta);

yc = y(i) + raio \* sin(theta);

%linhas de diametro

plot([x(i)-raio, x(i)+raio],[y(i), y(i)]); %horizontal

plot([x(i), x(i)],[y(i)-raio, y(i)+raio]); %vertical

%plot

plot(xc, yc);

plot(x(i), y(i), 'ok', 'markersize', 6);

%numero do círculo dentro

text(x(i)+0.2\*raio, y(i)+0.2\*raio, num2str(i), 'HorizontalAlignment', 'center', 'VerticalAlignment', 'middle', 'FontSize', 12);

end

disp(['Clique nos ', num2str(n), ' circulos na ordem certa:']);

[xclique, yclique] = ginput(n); %pega as coordenadas dos cliques

acertos = zeros(1,n); %cria o vetor do tamanho certo de uma vez

for i = 1:n

dx = xclique(i) - x;

dy = yclique(i) - y;

dist2 = dx.^2 + dy.^2;

[distanciocentromaisperto, indicecentro] = min(dist2); %pega qual o menor valor de distância até o centro, e o indice desse centro (qual i é ele)

if distanciocentromaisperto <= raio^2 && indicecentro == i

acertos(i) = 1;

endif

end

if sum(acertos) == n %soma de todos os elementos do vetor

disp('voce acertou');

else

disp('errou');

end

**exercicio2.1.m**

**exercicio2.2.m**

**exercicio2.3.m**

**exercicio2.4.m**

**exercicio3.1.m**

**exercicio4.1.m**

**exercicio4.2.m**

**exercicio5.1.m**

**exercicio5.2.m**

**exercicio5.3.m**

**exercicio5.4.m**

**exercicio6.1.m**

**exercicio6.2.m**

**exercicio6.3.m**

**exercicio6.4.m**

**exercicio7.1.m**

**exercicio7.2.m**

**exercicio7.3.m**