Table of Contents

Problem 1 abcd

```
clear all;
clc

C21 = [[0 0 1];[1 0 0];[0 1 0]]; % Given
myString1 = string(joshIsOnes(C21*C21' == eye(3) & C21'*C21 == eye(3) &
 det(C21) == 1)); % 1a

[a_v,phi] = joshRotM2PrincAxe(C21) % a_v and phi

phi_sym = sym(phi); % cast symbolic for readablility
a_v_sym = sym(a_v);

C21_star = joshPrincAxe2RotM(a_v,-phi);
C21_star = round(C21_star,15) % round off any errors near e-mach

myString2 = string(isequal(C21',C21_star)); % C21' == C21_star

C21_pound = joshPrincAxe2RotM(-a_v,-phi);
C21_pound = round(C21_pound,15)

myString3 = string(isequal(C21,C21_pound)); % C21 == C21_pound

disp("My workings for Problem 1 have the following results:")
disp("C21 is a rotation matrix: "+myString1)

disp("axis vector a is: ")
disp("   " + string(a_v_sym))
disp("rotation angle phi is: "+string(phi_sym))

disp("C21 transposed is = to C21*: "+myString2)
disp("C21 is = to C21#: " + myString3)
```

```
a_v =

   -0.5774
   -0.5774
   -0.5774


phi =
```

```
    2.0944


C21_star =

     0     1     0
     0     0     1
     1     0     0


C21_pound =

     0     0     1
     1     0     0
     0     1     0

My workings for Problem 1 have the following results:
C21 is a rotation matrix: true
axis vector a is:
    "    -3^(1/2)/3"
    "    -3^(1/2)/3"
    "    -3^(1/2)/3"

rotation angle phi is: (2*pi)/3
C21 transposed is = to C21*: true
C21 is = to C21#: true
```

Problem 2

```
clear all;

Cx = @(theta)...
    [[1 0 0];...
    [0 cos(theta) sin(theta)];...
    [0 -sin(theta) cos(theta)]];

Cy = @(theta)...
    [[cos(theta) 0 -sin(theta)];...
    [ 0 1 0];...
    [sin(theta) 0 cos(theta)]];

Cz = @(theta)...
    [[cos(theta) sin(theta) 0];...
    [-sin(theta) cos(theta) 0];...
    [0 0 1]];

syms t  [3 1] % x y and z thetas
assume(t,'real');

Cy1 = Cy(t(1)) % generate the individual rotation matrices
Cz1 = Cz(t(2))
Cx1 = Cx(t(3))
```

```
C21 = Cx1*Cz1*Cy1;

C21s1 = subs(C21,t(1),pi/2)
C21s2 = subs(C21,t(2),pi/2)
C21s3 = subs(C21,t(3),pi/2)

disp("My workings for Problem 2 have the following results:")
disp("C21 for a 2-3-1 rotation is given by: ")
disp("   "+string(C21))
disp("by analysis of the matrices resulting from substituting pi/2 for theta-x, theta-y, theta-z respectively, only theta-z = pi/2 is indeterminant")
```

```
Cy1 =

[cos(t1), 0, -sin(t1)]
[      0, 1,        0]
[sin(t1), 0,  cos(t1)]


Cz1 =

[ cos(t2), sin(t2), 0]
[-sin(t2), cos(t2), 0]
[       0,       0, 1]


Cx1 =

[1,        0,       0]
[0,  cos(t3), sin(t3)]
[0, -sin(t3), cos(t3)]


C21s1 =

[      0,          sin(t2),         -cos(t2)]
[sin(t3),  cos(t2)*cos(t3),  cos(t3)*sin(t2)]
[cos(t3), -cos(t2)*sin(t3), -sin(t2)*sin(t3)]


C21s2 =

[                                0, 1,                                 0]
[sin(t1)*sin(t3) - cos(t1)*cos(t3), 0, cos(t1)*sin(t3) + cos(t3)*sin(t1)]
[cos(t1)*sin(t3) + cos(t3)*sin(t1), 0, cos(t1)*cos(t3) - sin(t1)*sin(t3)]


C21s3 =

[cos(t1)*cos(t2),  sin(t2), -cos(t2)*sin(t1)]
[        sin(t1),        0,          cos(t1)]
[cos(t1)*sin(t2), -cos(t2), -sin(t1)*sin(t2)]
```

```
My workings for Problem 2 have the following results:
C21 for a 2-3-1 rotation is given by:
    "   cos(t1)*cos(t2)"      "   sin(t2)"              "   -cos(t2)*sin(t1)"
    "   sin(t1)*sin(t3)…"    "   cos(t2)*cos(t3)"     "   cos(t1)*sin(t3)…"
    "   cos(t3)*sin(t1)…"    "   -cos(t2)*sin(t3)"    "   cos(t1)*cos(t3)…"

by analysis of the matrices resulting from substituting pi/2 for theta-x,
 theta-y, theta-z respectively, only theta-z = pi/2 is indeterminant
```

Problem 3 part 1 b

```
clear all;

syms a [3 1]
assume(a,'real') % is vector of real numbers
assumeAlso(sqrt(sum(a.^2))==1) % is unit vector
% assumptions(a)
ax = joshCross(a)
LHS = ax*ax*ax
RHS = -ax
myString = string(joshIsOnes(isAlways(RHS == LHS))); % returns a matrix of
 logical and checks each individual LHS value is always the corresponding
 value on RHS

disp("My workings for Problem 3 part 1 b have the following results:")
disp("axaxax is = to -ax: "+myString)
```

```
ax =

[  0, -a3,  a2]
[ a3,   0, -a1]
[-a2,  a1,   0]


LHS =

[                            0,   a3*a1^2 + a3*(a2^2 + a3^2), - a2*a1^2 -
 a2*(a2^2 + a3^2)]
[- a3*a2^2 - a3*(a1^2 + a3^2),                            0,   a1*a2^2 +
 a1*(a1^2 + a3^2)]
[  a2*a3^2 + a2*(a1^2 + a2^2), - a1*a3^2 - a1*(a1^2 + a2^2),
          0]


RHS =

[  0,  a3, -a2]
[-a3,   0,  a1]
[ a2, -a1,   0]

My workings for Problem 3 part 1 b have the following results:
```

```
axaxax is = to -ax: true
```

Problem 3 part 2

```
clear all;

Cx = @(theta)...
    [[1 0 0];...
    [0 cosd(theta) sind(theta)];...
    [0 -sind(theta) cosd(theta)]];

syms A B

LHS = Cx(A+B)
RHS = Cx(A)*Cx(B)
myString = string(joshIsOnes(isAlways(RHS == LHS)));
disp("My workings for Problem 3 part 2 have the following results:")
disp("Cx(A+B) is = to Cx(A)+Cx(B): "+myString)
```

```
LHS =

[1,                      0,                      0]
[0,  cos((pi*(A + B))/180), sin((pi*(A + B))/180)]
[0, -sin((pi*(A + B))/180), cos((pi*(A + B))/180)]


RHS =

[1,                                                                   0,
                                                          0]
[0,   cos((pi*A)/180)*cos((pi*B)/180) - sin((pi*A)/180)*sin((pi*B)/180),
 cos((pi*A)/180)*sin((pi*B)/180) + cos((pi*B)/180)*sin((pi*A)/180)]
[0, - cos((pi*A)/180)*sin((pi*B)/180) - cos((pi*B)/180)*sin((pi*A)/180),
 cos((pi*A)/180)*cos((pi*B)/180) - sin((pi*A)/180)*sin((pi*B)/180)]

My workings for Problem 3 part 2 have the following results:
Cx(A+B) is = to Cx(A)+Cx(B): true
```

Published with MATLAB® R2022a

```
function [a_v,phi] = joshRotM2PrincAxe(C)
arguments
    C (:,:) {mustBeReal, mustBeNumeric}
end

[m,n] = size(C);

if m ~= n % matrix must be square
    throw(MException("joshRotM2PrincAxe:invalidInput","Dimensions of C21 must
 match."))
end
clear m

if ~(round(C*C',14) == eye(n) & round(C'*C,14) == eye(n) & round(det(C),14) ==
 1) % checks that M is a rotation matrix, round so that an error near e-mach
 will not cause a failure
    throw(MException("joshPrincAxe:invalidInput","Matrix is not a rotation
 matrix."))
end
    phi = acos((trace(C)-1)/2); % calculates phi in terms of C
    if abs(phi - pi)<1e-14 % checks if there is a non unique solution
        warning("answer may not be unique")
    end
    a_v(1) = (C(2,3)-C(3,2))/(2*sin(phi)); % formula for components of a in
 terms of phi and C
    a_v(2) = (C(3,1)-C(1,3))/(2*sin(phi));
    a_v(3) = (C(1,2)-C(2,1))/(2*sin(phi));
    a_v = a_v';
end
```

Published with MATLAB® R2022a

```
function [isOnes] = joshIsOnes(M)
[m,n] = size(M);
isOnes = true;
for i = 1:m
    for j = 1:n
        if M(i,j) ~= 1
            isOnes = false;
        end
    end
end
end
```

Published with MATLAB® R2022a

```
function mx = joshCross(m)
arguments
    m (3,1)
end
if isa(m,"double")
    mx = zeros(3);
elseif isa(m,"sym")
    syms mx [3 3]
else
    throw(MException("joshCross:invalidInput","m must be type sym or double"))
end
    for i = 1:3
        mx(i,i) = 0;
    end
    mx(1,2) = -m(3);
    mx(1,3) =  m(2);
    mx(2,3) = -m(1);

    mx(2,1) =  m(3);
    mx(3,1) = -m(2);
    mx(3,2) =  m(1);
end
```

Published with MATLAB® R2022a

```
function [C] = joshPrincAxe2RotM(a,phi)
arguments
    a (1,3) {mustBeReal, mustBeNumeric}
    phi (1,1) {mustBeReal, mustBeNumeric}
end
if abs(norm(a) - 1) > 1e-14 % checks if a is a unit vector, rounds so that an
 error near e-mach will not cause a failure
    throw(MException("joshPrincAxe2RotM:invalidInput","a is not a unit
 vector"))
end
[n,m] = size(a);
if (n == 3 & m == 1)|(n == 1 & m == 3) % a must be a 1x3 or 3x1
    if m == 3
        a = a'; % if a is a horizontal vector it will be transposed to
 vertical
    end
else % a must have 3 components
    throw(MException("joshPrincAxe2RotM:invalidInput","a is not a 1x3 or
 3x1"))
end

ax = joshCross(a); % a-"cross", returns a symbolic type
ax = double(ax); % ax cast to double from symbolic

x=ax(1); % short hands for readablility
y=ax(2);
z=ax(3);
c=cos(phi);
s=sin(phi);
Cs = 1-c; % shortHand usually called C

C = c*eye(3)+Cs*(a*a')-s*ax; % calculates C in terms of ax and phi

end
```

Published with MATLAB® R2022a

HW2

Problem 3 1a)

Show $\vec{a} \times [\vec{a} \times (\vec{a} \times \vec{b})]$ $|\vec{a}| = 1$ $= -\vec{a} \times \vec{b}$

$\vec{a} \cdot \vec{a} = |\vec{a}|^2$

$= \vec{a} \times [(\vec{b} \cdot \vec{a})\vec{a} - (\vec{a} \cdot \vec{a})\vec{b}]$

$= \vec{a} \times [(\vec{b} \cdot \vec{a})\vec{a} - \vec{b}]$

distributive

$= \vec{a} \times [(\vec{b} \cdot \vec{a})\vec{a}] + (\vec{a} \times -\vec{b}) = -\vec{a} \times \vec{b}$

Let $s = \vec{b} \cdot \vec{a} \in \mathbb{R}^1$

$\Rightarrow \vec{a} \times (s\vec{a}) + (\vec{a} \times -\vec{b}) = -\vec{a} \times \vec{b}$

let $\theta =$ the angle between $\vec{a}$; $s\vec{a}$

$\Rightarrow \theta = 0$, $\Rightarrow |\vec{a} \times (s\vec{a})| = |\vec{a}||s\vec{a}| \sin\theta = 0$

$\Rightarrow \vec{0} + (\vec{a} \times -\vec{b}) = (\vec{a} \times -\vec{b}) \Rightarrow \vec{a} \times -\vec{b} = -\vec{a} \times \vec{b}$

$c = -1$, use $\vec{a} \times c\vec{b} = c\vec{a} \times \vec{b} = c(\vec{a} \times \vec{b})$

$\boxed{-\vec{a} \times \vec{b} = -\vec{a} \times \vec{b}}$