

---

```
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*