Three Residue Loop Closure in Proteins: A New Kinematic Method Reveals a Locus of Connected Loop Conformations

Supplementary Document: MATLAB Code for Comparison of the Computation Time with the Polynomial Method

Ali Nekouzadeh and Yoram Rudy

Cardiac Bioelectricity and Arrhythmia Center and Biomedical Engineering Department, Washington University in St. Louis

MATLAB Code

```
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%  New Kinematic Method  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants
k=1;
Csi3=Csi1ma;
a=1.1;
b=3.6;
S70=sind(70.5);
C70=cosd(70.5);

% Locations and direction is assumed in 1' system (constant psi0 and phi1)
Ang1V=0.25:0.5:pi; % different values for theta_d
Ang2V=0.5:0.5:2*pi; % different values for gamma_d
rV=2:1:9; % different values for r
tetV=0.25:0.5:pi; % different values for theta_r
l1=length(tetV)*length(rV)*length(Ang1V)*length(Ang2V); % Total number of
combinations

for bb=1:10 % Computing the computation time 10 times
    cpt=cputime; % starting to measure the computation time
    for Ang1=Ang1V;
        for Ang2=Ang2V
            Rs=[sin(Ang1)*cos(Ang2);sin(Ang1)*sin(Ang2);cos(Ang1)];
            for r=rV
                for tet=tetV
                    Loc=[r*sin(tet) ; 0 ; r*cos(tet)];
                    % vectors to keep solutions for psi1, psi2, psi3, alpha
and beta angles at each combination
                    si1SV=[];
                    si2SV=[];
                    si3SV=[];
                    alfSV=[];
                    betSV=[];
                    % analytical solutions for alpha and beta
                    Cb=(C70^2-Rs(3))/S70^2;
                    if abs(Cb)<=1
                        C=Rs(1)/(S70*sqrt(1-Cb^2+C70^2*(1+Cb)^2));
                        if abs(C)<=1
                            if Rs(2)>=0
                                kk=2;
                            else
                                kk=1;
                            end
                            bet=acos(Cb);
                            B=-sin(bet)*S70;
                            A=S70*C70*(1+Cb);
                            % numerical solution for one set of alpha and
beta
                            alf=(-1)^kk*acos(C)+atan(B/A);
                            Ds=[C70*cos(alf) C70*sin(alf) -S70 ; -sin(alf)
cos(alf)  0 ; S70*cos(alf) S70*sin(alf) C70]*Loc;
```

```
                            d1=Ds(3)/a/S70-b*(1+2*C70)/a/S70;
                            if abs(d1)<=2
                                si1ma=[acosd(max([(-1+d1) -
1])):1:acosd(min([(1+d1) 1])) acosd(min([(1+d1) 1]))]; % possible range of
(psi1-alpha)
                                Csi1ma=cosd(si1ma);
                                Csi3=Csi1ma-d1;

                                % first of the four options for the sign
of(psi1-alpha) and psi3
                                Ssi1ma=sqrt(1-Csi1ma.^2);
                                Ssi3=sqrt(1-Csi3.^2);
                                Csi2=Ds(1)/a-b*S70*cos(bet)/a-C70*Csi1ma-
C70*cos(bet)*Csi3+sin(bet)*Ssi3+b*S70/a;
                                Ssi2=-Ssi1ma-C70*sin(bet)*Csi3-
cos(bet)*Ssi3+Ds(2)/a-b*S70*sin(bet)/a;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f))); % finding the zero
crossing (psi1-alpha) angles

                                % linear interpolation for more precise
caclculation of angles
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % second of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2-2*sin(bet)*Ssi3;
                                Ssi2=Ssi2+2*cos(bet)*Ssi3;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;-acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % third of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2+2*C70*Csi1ma;
                                Ssi2=Ssi2+2*Ssi1ma;
```

```
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;-acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;-acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % fourth of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2+2*sin(bet)*Ssi3;
                                Ssi2=Ssi2-2*cos(bet)*Ssi3;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;-acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end
                            end

                            % numerical solution for other set of alpha and
beta
                            alf=(-1)^kk*acos(C)-atan(B/A);
                            Ds=[C70*cos(alf) C70*sin(alf) -S70 ; -sin(alf)
cos(alf)  0 ; S70*cos(alf) S70*sin(alf) C70]*Loc;
                            d1=Ds(3)/a/S70-b*(1+2*C70)/a/S70;
                            if abs(d1)<=2
                                si1ma=[acosd(max([(-1+d1) -
1])):5:acosd(min([(1+d1) 1])) acosd(min([(1+d1) 1]))];
                                Csi1ma=cosd(si1ma);
                                Csi3=Csi1ma-d1;

                                % first of the four options for the sign
of(psi1-alpha) and psi3
                                Ssi1ma=sqrt(1-Csi1ma.^2);
                                Ssi3=sqrt(1-Csi3.^2);
                                Csi2=Ds(1)/a-b*S70*cos(bet)/a-C70*Csi1ma-
C70*cos(bet)*Csi3+sin(bet)*Ssi3+b*S70/a;
                                Ssi2=-Ssi1ma-C70*sin(bet)*Csi3-
cos(bet)*Ssi3+Ds(2)/a-b*S70*sin(bet)/a;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
```

```
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % second of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2-2*sin(bet)*Ssi3;
                                Ssi2=Ssi2+2*cos(bet)*Ssi3;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;-acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % third of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2+2*C70*Csi1ma;
                                Ssi2=Ssi2+2*Ssi1ma;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
                                    si1SV=[si1SV;-acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;-acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end

                                % fourth of the four options for the sign
of(psi1-alpha) and psi3
                                Csi2=Csi2+2*sin(bet)*Ssi3;
                                Ssi2=Ssi2-2*cos(bet)*Ssi3;
                                f=Csi2.^2+Ssi2.^2-1;
                                IX=find(diff(sign(f)));
                                for m=IX
                                    p=abs(f(m))/(abs(f(m))+abs(f(m+1)));
```

```
                                    si1SV=[si1SV;-acos((1-
p)*Csi1ma(m)+p*Csi1ma(m+1))+alf];
                                    si2SV=[si2SV;acos((1-
p)*Csi2(m)+p*Csi2(m+1))*sign((1-p)*Ssi2(m)+p*Ssi2(m+1))];
                                    si3SV=[si3SV;acos((1-
p)*Csi3(m)+p*Csi3(m+1))];
                                    alfSV=[alfSV;alf];
                                    betSV=[betSV;bet];
                                end
                            end
                        end
                    end
                end
            end
        end

    tpl1(bb)=(cputime-cpt)/l1; % computation time for one trial (of 10
trials)
end

%%%%%%%%%%%%%%%%%%%%%%%%   Polynomial Method    %%%%%%%%%%%%%%%%%%%%%%%%%%
% constants
del=180*ones(1,3);
tet=109*ones(1,3);
Ctet=cosd(tet);

et1=17;
xi1=17;
et2=17;
xi2=17;
et3V=0:4:60; % different values of eta_3
xi3V=0:4:56; % different values of xi_3
alf1V=105:5:175; % different values of alpha_1
l2=length(et3V)*length(xi3V)*length(alf1V); % Total number of combinations

for bb=1:10 % Computing the computation time 10 times
    cpt=cputime; % starting to measure the computation time
    for et3=et3V
        for xi3=xi3V
            for alf1=alf1V
                alf3=alf1;
                alf2=360-2*alf1;

                alf=[alf1 alf2 alf3];
                et=[et1 et2 et3];
                xi=[xi3 xi1 xi2];

                p=zeros(3,3,3);
                S=zeros(6,6,5);
                q=zeros(5,5);

                % vectors to keep solutions for tau1, tau2, tau3 at each
combination
                Tau1SV=[];
```

```
Tau2SV=[];
Tau3SV=[];

Sxi=sind(xi);
Cxi=cosd(xi);
Same=sind(alf-et);
Sape=sind(alf+et);
Came=cosd(alf-et);
Cape=cosd(alf+et);
% determining the p coefficients
p(3,3,1:3)=-Ctet-cosd(xi).*Came+Sxi.*Same;
p(3,1,1:3)=-Ctet-cosd(xi).*Cape+Sxi.*Sape;
p(2,2,1:3)=-4*Sxi.*sind(et);
p(1,3,1:3)=-Ctet-Cxi.*Came-Sxi.*Same;
p(1,1,1:3)=-Ctet-Cxi.*Cape-Sxi.*Sape;
% determining coefficients of the elements of L matrices
L0=sum(p(1,:,1));
L1=sum(p(2,:,1));
L2=sum(p(3,:,1));
% determining coefficients of the elements of M matrices
M0=sum(p(1,:,2));
M1=sum(p(2,:,2));
M2=sum(p(3,:,2));
% determining the elements of Q matrix
q(1,5)=L2^2*M0^2;
q(5,1)=L0^2*M2^2;
q(3,3)=-2*L0*L2*M0*M2+L2*M1^2*L0+M2*L1^2*M0;
q(2,4)=-L2*L1*M1*M0;
q(4,2)=-L1*L0*M2*M1;
% determining the elements of S matrices
S(1,1,1:3)=p(3,:,3);
S(1,2,1:3)=p(2,:,3);
S(1,3,1:3)=p(1,:,3);
S(2,2,1:3)=p(3,:,3);
S(2,3,1:3)=p(2,:,3);
S(2,4,1:3)=p(1,:,3);
S(3,3,1:3)=p(3,:,3);
S(3,4,1:3)=p(2,:,3);
S(3,5,1:3)=p(1,:,3);
S(4,4,1:3)=p(3,:,3);
S(4,5,1:3)=p(2,:,3);
S(4,6,1:3)=p(1,:,3);
S(5,1,:)=q(5,:);
S(5,2,:)=q(4,:);
S(5,3,:)=q(3,:);
S(5,4,:)=q(2,:);
S(5,5,:)=q(1,:);
S(6,1,:)=q(5,:);
S(6,1,:)=q(4,:);
S(6,1,:)=q(3,:);
S(6,1,:)=q(2,:);
S(6,1,:)=q(1,:);
% using QZ factorization to determine the solutions of
characteristic equation for u3
A=[[zeros(6,6) eye(6) zeros(6,6) zeros(6,6)];[zeros(6,6)
zeros(6,6) eye(6) zeros(6,6)];[zeros(6,6) zeros(6,6) zeros(6,6) eye(6)];[-
S(:,:,1) -S(:,:,2) -S(:,:,3) -S(:,:,4)]];
```

```
                B=[[eye(6) zeros(6,6) zeros(6,6) zeros(6,6)];[zeros(6,6)
eye(6) zeros(6,6) zeros(6,6)];[zeros(6,6) zeros(6,6) eye(6)
zeros(6,6)];[zeros(6,6) zeros(6,6) zeros(6,6) -S(:,:,5)]];
                E=eig(A,B);
                E=E(real(E)~=0 & real(E)~=Inf & real(E)~=-Inf & imag(E)==0);

                % determining the tau angles from the known values of u3
                for i=1:length(E)
                    u3=E(i);
                    Tau3SV=[Tau3SV;2*atan(u3)];

                    u3V3=[1 u3 u3^2];
                    u3V5=[1 u3 u3^2 u3^3 u3^4];
                    N2=sum(p(3,:,3).*u3V3);
                    N1=sum(p(2,:,3).*u3V3);
                    N0=sum(p(1,:,3).*u3V3);
                    Q4=sum(q(5,:).*u3V5);
                    Q3=sum(q(4,:).*u3V5);
                    Q2=sum(q(3,:).*u3V5);
                    Q1=sum(q(2,:).*u3V5);
                    Q0=sum(q(1,:).*u3V5);
                    u2=det([N2 N1 N0 0 0 ; 0 N2 N1 N0 0 ; 0 0 N2 N1 0 ; 0 0 0
N2 -N0 ; 0 Q4 Q3 Q2 -Q0])/det([N2 N1 N0 0 0 ; 0 N2 N1 N0 0 ; 0 0 N2 N1 N0 ; 0
0 0 N2 N1 ; 0 Q4 Q3 Q2 Q1]);
                    Tau2SV=[Tau2SV;2*atan(u2)];

                    u1=det([L2*u3^2 L1*u3 0 ; 0 L2*u3^2 -L0 ; 0 M2*u2^2 -
M0])/det([L2*u3^2 L1*u3 L0 ; 0 L2*u3^2 L1*u3 ; 0 M2*u2^2 M1*u2]);
                    Tau1SV=[Tau1SV;2*atan(u1)];
                end
            end
        end
    end
    tpl2(bb)=(cputime-cpt)/l2; % computation time for one trial (of 10
trials)
end


mean(tpl1) % average computation time in new kinematic method
mean(tpl2) % average computation time in polynomial method

std(tpl1)
std(tpl2)
```