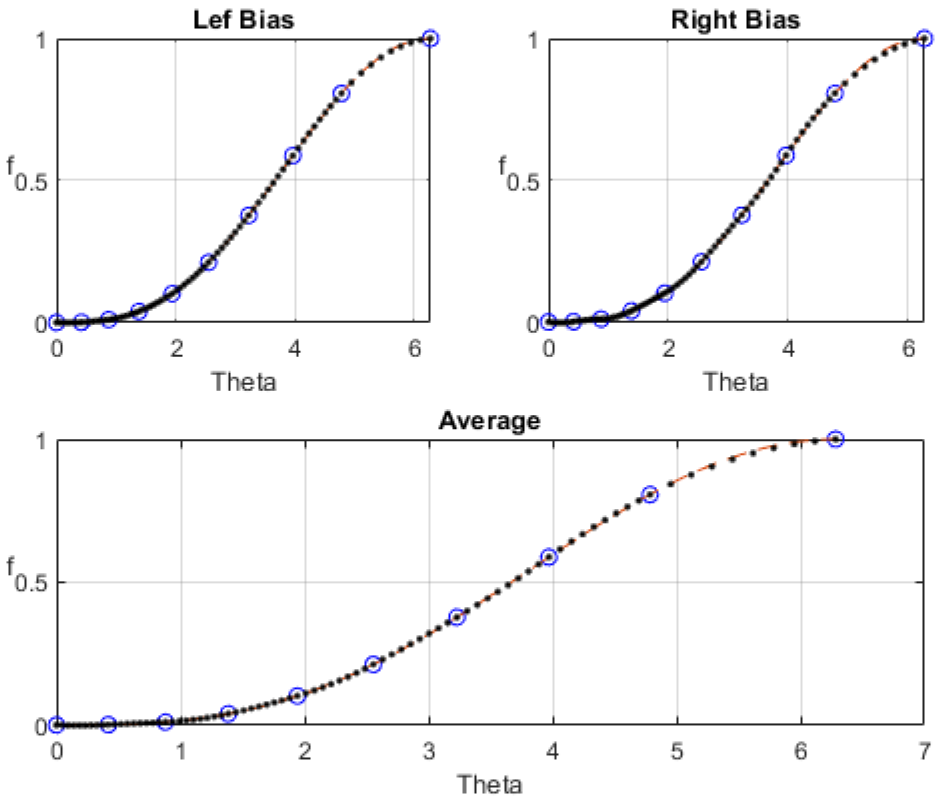
```matlab
clear, clc;
close all;

%case 1 data points and grid condition
n = 10;
del = .4186;
k = 1.1;

%generating grid
x = ones(1,n);
x(1) = 0;
x(n) = 2*pi();
for i = 1:n-2
    x(i+1) = x(i) + del*(k^(i-1));
end

%generating grid range points (change orginal eqn here)
y = sin(x./4).^3;

%exact original fncs
xexact = linspace(0,2*pi(),100);
yexact = sin(xexact./4).^3;

%running methods with inputs from case 1 to get left bias and right bias
%splines
sl = LQuadSpline(x,y,n);
sr = RQuadSpline(x,y,n);

% creating average quadratic splines by averaging x and y values of left
% and right bias splines
sa = cell(n-1,1);
for i = 1:n-1
    sa{i} = (sl{i}+sr{i})./2;
end

%Plotting left/right bias splines and the average
figure(1);
subplot(2,2,1)
QuadPlot(x,y,xexact,yexact,sl,n,'Lef Bias');
subplot(2,2,2)
QuadPlot(x,y,xexact,yexact,sr,n, 'Right Bias');
subplot(2,2,[3,4])
QuadPlot(x,y,xexact,yexact,sa,n, 'Average');

%Solving for left bias quadratic splines
function splines = LQuadSpline(x,y,n)
    %Setting up matrix system to solve for k1
    A = [(x(2)-x(1)),((x(2)-x(1))^2);(x(3)-x(1)),((x(3)-x(1))^2)];
    b = [y(2);y(3)];
    k = ones(n-1,1);
    m = ones(n-1,1);
    k(1) = cramer(A,b,1);
    %solving for m and the rest of k from left to right
    for i = 1:n-1
        m(i) = ((y(i+1)-y(i))-k(i)*(x(i+1)-x(i)))/((x(i+1)-x(i))^2);
        k(i+1) = k(i) + 2*m(i)*(x(i+1)-x(i));
    end

    %setting up 10 points for each spline and solve for splines
    splines = cell(n-1,1);
    for i = 1:n-1
        xspline = linspace(x(i),x(i+1),10);
        yspline = y(i)+k(i)*(xspline-x(i))+m(i)*(xspline-x(i)).^2;
        splines{i} = [xspline;yspline];
    end
end

function splines = RQuadSpline(x,y,n)
    %Setting up matrix system to solve for kn
    A = [(x(n-1)-x(n)),((x(n-1)-x(n))^2);(x(n-2)-x(n)),((x(n-2)-x(n))^2)];
    b = [(y(n-1)-y(n));(y(n-2)-y(n))];
    k = ones(n-1,1);
    m = ones(n-1,1);
    k(n) = cramer(A,b,1);

    %Solving for the rest of m and k from right to left
    for i = n:-1:2
        m(i) = ((y(i-1)-y(i))-k(i)*(x(i-1)-x(i)))/((x(i-1)-x(i))^2);
        k(i-1) = k(i) + 2*m(i)*(x(i-1)-x(i));
    end

    %set up 10 points per subdivision and solve for right bias splines
    splines = cell(n-1,1);
    for i = 2:n
        xspline = linspace(x(i-1),x(i),10);
        yspline = y(i)+k(i)*(xspline-x(i))+m(i)*(xspline-x(i)).^2;
        splines{i-1} = [xspline;yspline];
    end
end

% plot x-y data points
% plot exact function
% plot splines
function QuadPlot(x,y,xexact,yexact,s,n,str)
    plot(x, y, 'bo',xexact,yexact, '--')
    hold on; grid on;
    for i = 1:n-1
        plot(s{i}(1,:),s{i}(2,:),'.k')
    end
    title(str);
    xlabel('Theta');
    ylabel('f','Rotation',0);
end

%take matrix a,b, and column values then use cramer's rule
```

```
function cram = cramer(a,b,col)
    deterA = (a(1,1).*a(2,2))-(a(1,2).*a(2,1));
    ai = a;
    ai(:,col) = b;
    deterAI = (ai(1,1).*ai(2,2))-(ai(1,2).*ai(2,1));
    cram = deterAI/deterA;
end
```



Published with MATLAB® R2021b