Contents

B-Spline Approximation of x(t)

Problem 7.4

```
close all;
clear

for L = 1:4
```

Setup

```
    N = 20;
    W = [-pi:0.0001:pi];
    t = [-100:0.0001:100];
    h_L = zeros(1, 2*N+1);
    h_L_half = zeros(1, N+1);
    H_L = zeros(1, length(W));
    x = @(t) 0.*(t<=0) + 0.5 .* ((t>0) .* (t<=10)) - sin(pi*t/10) .* ((t>10) .* (t<=20)) + 0.
*(t>20);
    figure;
    subplot(2, 2, 1);
    plot(t, x(t));
    title('Original Signal $$x(t)$$', 'Interpreter', 'latex');
    if L == 1
        b = @(t1) bspline1(t1);
    elseif L == 2
        b = @(t1) bspline2(t1);
    elseif L == 3
        b = @(t1) bspline3(t1);
    elseif L == 4
        b = @(t1) bspline4(t1);
    end
```

Computing h_L

Finding H_L

```
    for i = 1:length(W)
        H_L(i) = 1 / G_L(W(i), L);
```

```
    end

    for i = 1:N
        h_L_half(i) = 0;
        % n = i - (N+1);
        h_L_half(i) = sum(H_L .* cos(W*(i-1)));
        h_L_half(i) = h_L_half(i);
    end


    h_L = [fliplr(h_L_half), h_L_half(2:end)];
    h_L = h_L / (length(W));
    % figure;
    subplot(2, 2, 2);
    plot([-N:N], h_L);
    title('$$h_L[n]$$', 'Interpreter', 'latex')
    xlabel('n');
    ylabel('h_L[n]');
```

Finding alpha[n]

```
    prod_f = @(t1, n, l) (x(t1) .* b(t1-n-l));

    n_r = 40;
    l_r = 100;
    al = zeros(1, 2*n_r+1);
    for n = -n_r:n_r
        for l = 1:length(h_L)
            al(n+n_r+1) = al(n+n_r+1)  + h_L(l) * integral(@(t1)(prod_f(t1, n, l - (N+1))), -
100, 100);
        end
    end
    % figure;
    subplot(2,2,3);
    stem([-n_r:n_r], al);
    title('Basis coefficients $$(\alpha_n)$$', 'Interpreter', 'latex');
    ylabel('\alpha_n');
    xlabel('n');
```

Estimating with B-Splines and Dual Basis

```
    y = zeros(1, length(t));
    for n = -n_r:n_r
        y = y + al(n+n_r+1) * b(t-n);
    end
    %figure;
    subplot(2,2,4)
    hold on;
    plot(t, y);
    title('Original Signal: $$x(t)$$, Estimated Signal $$y(t)$$', 'Interpreter', 'latex');
    plot(t, x(t));
    ylabel('y(t), x(t)');
    xlabel('t');
```

Computing Error

```
    err = 0;
    dt = (t(2)-t(1));
    for i = 1:length(t)
        err = err + (abs(y(i) - x(t(i))) ^ 2) * dt;
    end
    disp(['Error for L = ' num2str(L) ' ' num2str(err)]);
```

```
end
```

Helper Function for Finding G_L(e^(jw))

Finding G_L

```
function Gw = G_L(w, L)
    Gw = 0;
    K = 50;
    for k = -K:K
        Gw = Gw + ((sin((w/2) + pi*k)) / ((w/2) + pi*k)) ^ (2*L + 2);
    end
end
```

Published with MATLAB® R2017a