

Problem1

November 5, 2016

```
In [8]: using Plots
        using ApproxFun
```

1 Part 1

```
In [2]: xi = [0; 20; 40; 60; 80; 100]
        yi = [76.0; 105.7; 131.7; 179.3; 226.5; 281.4]
```

```
scatter(xi,yi)
```

[Plots.jl] Initializing backend: pyplot

```
In [3]: """
        'lagrange_interp': lagrange_interp(x, xi, yi)
        y = lagrange_interp(x,xi,yi) uses the data from xi,yi to build the
        Lagrange interpolant and then evaluates that interpolant at the points
        x, returning the interpolated values y.
        """
        function lagrange_interp(x, xi, yi)
            y = zeros(size(x))
            for i=1:length(yi)
                elli = ones(size(x))
                for j=1:length(xi)
                    if i==j continue; end
                    elli = elli.*(x-xi[j])/(xi[i] - xi[j])
                end
                y = y + yi[i]*elli
            end
            return y
        end
        ## Lagrange form
        xx = linspace(0,120,1000)
        yyl = lagrange_interp(xx,xi,yi);
        plot(xx,yyl); scatter!(xi,yi)
```

$$p(x) = 76 \frac{(x-20)(x-40)}{(-20)(-40)} + 105.7 \frac{x(x-40)}{20(20-40)} + 131.7 \frac{x(x-20)}{40(40-20)}$$

```
In [4]: p1 = x -> 76.0*((x)-(20))*((x)-(40))/((-20)*(-40))
        p2 = x -> 105.7*(x)*((x)-(40))/((20)*(20-40))
        p3 = x -> 131.7*(x)*((x)-(20))/((40)*(40-20))
        f = x -> p1(x)+p2(x)+p3(x)
```

```

n=200
yl=zeros(n,1)
tvals = linspace(0,40,n)
for i=1:n
    yl[i] = f(tvals[i])
end
plot(tvals, yl)

```

2 Part 2

Direct computation of Lagrange interpolant

```

In [5]: year_to_ms = 365*24*60*60*1000
p1 = x -> 76.0*((x*year_to_ms)-(20*year_to_ms))*((x*year_to_ms)-(40*year_to_ms))/((-20*year_to_ms)*(40*year_to_ms))
p2 = x -> 105.7*(x*year_to_ms)*((x*year_to_ms)-(40*year_to_ms))/((20*year_to_ms)*(20*year_to_ms))
p3 = x -> 131.7*(x*year_to_ms)*((x*year_to_ms)-(20*year_to_ms))/((40*year_to_ms)*(40*year_to_ms))
f = x -> p1(x)+p2(x)+p3(x)

n=50
yl=zeros(n,1)
tvals = linspace(0,40*year_to_ms,n)
for i=1:n
    yl[i] = f(tvals[i])
end
plot(tvals, yl)

```

LoadError: UndefVarError: y2ms not defined
while loading In[5], in expression starting on line 9

Direct Barycentric interpolant

```

In [6]: y2ms = 365*24*60*60*1000
w0 = 1/((-20*y2ms)*(-40*y2ms))
w1 = 1/((20*y2ms)*(20*y2ms-40*y2ms))
w2 = 1/((40*y2ms)*(40*y2ms*20*y2ms))

p = x -> (w0*yi[1]/(x) + w1*yi[2]/(x-(20*y2ms)) + w2*yi[3]/(x-(40*y2ms))) / (w0/(x) + w1/(x-(20*y2ms)) + w2/(x-(40*y2ms)))

n=50
yb=zeros(n,1)
tvals = linspace(0,40*y2ms,n)
for i=1:n
    yb[i] = p(tvals[i])
end

plot(tvals, yb)

```

3 Part 3

```

In [7]: function barylag(x,y,xx)
    # direct port of

```

```

# http://www.mathworks.com/matlabcentral/fileexchange/...
# 4478-barycentric-lagrange-interpolating-polynomials-...
# and-lebesgue-constant/content/barylag.m
# to Julia. Not that better implmentations in Julia are possible.
M = length(x)
N = length(xx)
@assert M == length(y)
X = repmat(x,1,M)
W = repmat(1./prod(X-X'+eye(M),1),N,1)
xdist=repmat(xx,1,M)-repmat(x',N,1)
fixi,fixj = findnz(xdist.==0)
H=W./xdist
p=vec((H*y)./sum(H,2))
p[fixi] = y[fixj]
return p
end
ybl = barylag(xi[1:3]*y2ms, yi[1:3],linspace(0,40*y2ms,50))
plot(tvals, ybl)

```

In []: