```
function [ xplot, yplot1, yplot2, x, y, yplot ] = PlotCubicSpline( h )
%PlotCubicSpline Plot function and its cubic spline interpolations
%   Input = meshsize h, Output = values of f, s1, s2 at xplot

f = inline('exp(x).*cos(2*pi*x)');
fp = inline('exp(x).*cos(2*pi*x) - 2*pi*exp(x).*sin(2*pi*x)');

x = 0:h:1;
y = f(x);

alpha1 = HermiteEnd( x, h, f, fp );
alpha2  = NaturalEnd( x, h, f );

hplot = 0.01;
xplot = 0-h:hplot:1+h;
yplot = f(xplot);
yplot1 = zeros(size(xplot));
yplot2 = zeros(size(xplot));
N = length(x);
for r=-3:N-2
    ytemp = CubicBSline( r, h, xplot );
    yplot1 = yplot1 + alpha1(r+4)*ytemp;
    yplot2 = yplot2 + alpha2(r+4)*ytemp;
end

plot(xplot,yplot,'-k', xplot,yplot1,'.-r', xplot,yplot2,'--b',x,y,'ok');
legend('exact','Hermite','Natural','data');
legend('Location', 'SouthWest');

end
```

```
function [ alphas ] = HermiteEnd( x, h, f, fp )
%HermiteEnd Compute \alpha, which is coeff. for a cubic spline s(x)
%   Input: x = vector of [x_0, x_1, ..., x_n]

N = length(x);

b = cat(2,fp(x(1)),f(x),fp(x(end)));
b = b';

A = (2/3)*diag(ones(N+2,1)) + (1/6)*diag(ones(N+1,1),1) + (1/6)*diag(ones(N+1,1),-1);
A(1,1) = -1/(2*h);
A(1,3) = 1/(2*h);
A(end,end-2) = -1/(2*h);
A(end,end) = 1/(2*h);

alphas = A\b;

end
```

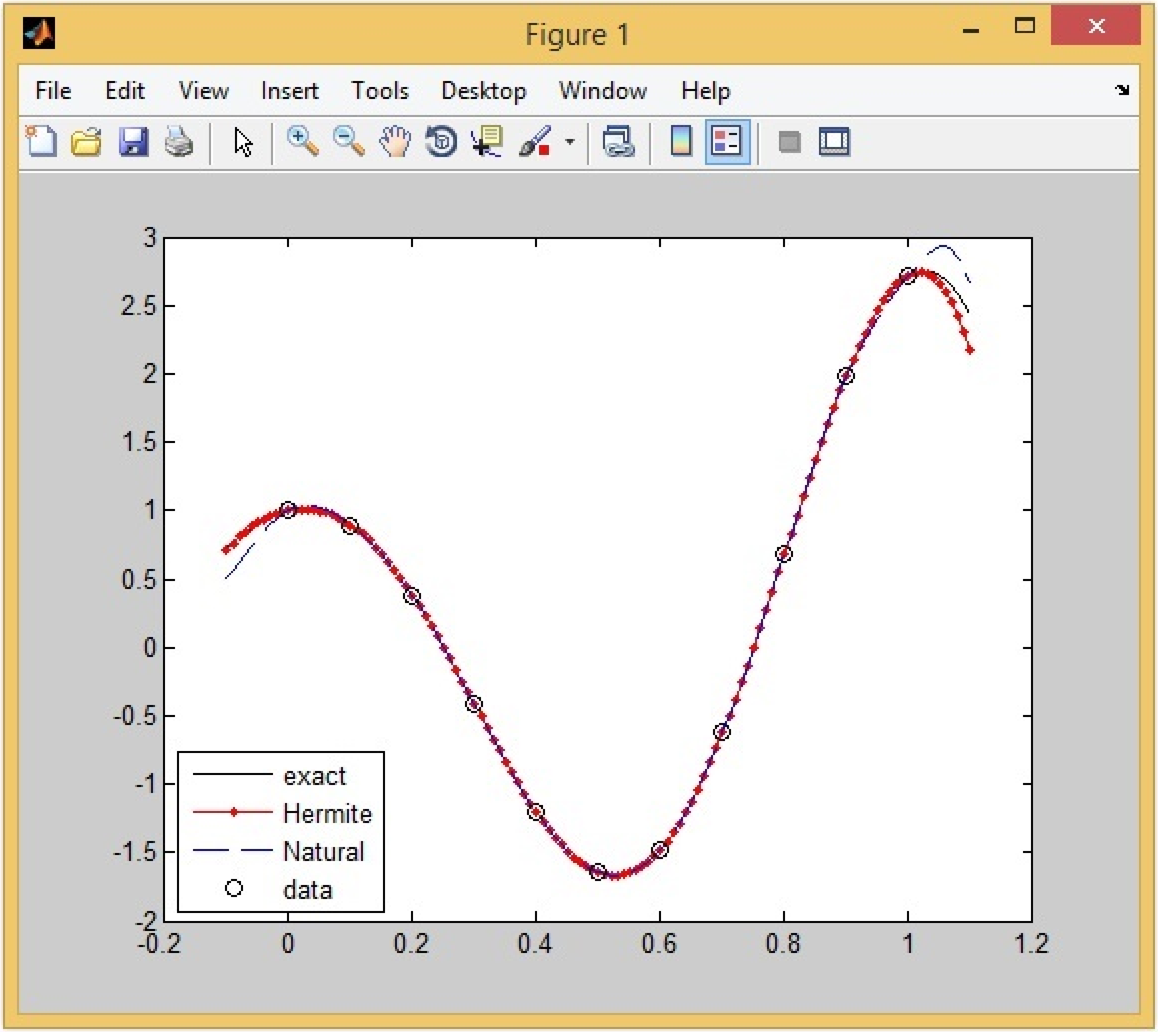
```
function [ alphas ] = NaturalEnd( x, h, f )
%HermiteEnd Compute \alpha, which is coeff. for a cubic spline s(x)
%   Input: x = vector of [x_0, x_1, ..., x_n]

N = length(x);

b = cat(2,0,f(x),0);
b = b';

A = (2/3)*diag(ones(N+2,1)) + (1/6)*diag(ones(N+1,1),1) + (1/6)*diag(ones(N+1,1),-1);
A(1,1) = 1/(h^2);
A(1,1) = -2/(h^2);
A(1,3) = 1/(h^2);
A(end,end-2) = 1/(h^2);
A(end,end-1) = -2/(h^2);
A(end,end) = 1/(h^2);

alphas = A\b;

end
```

```
function [ y ] = CubicBSline( ind, h, x )
%CubicBSline Compute the value B_i(x) where B_i starts at x_i = ih
%   Input: ind = index, h = mesh size, x = vector x
%   Output: y = B_ind(x)

xi = ind*h;
xip1 = (ind+1)*h;
xip2 = (ind+2)*h;
xip3 = (ind+3)*h;
xip4 = (ind+4)*h;

p1 = (x-xi).^3/(6*h^3);
p2 = 1/6 + (x-xip1)/(2*h) + (x-xip1).^2/(2*h^2) - (x-xip1).^3/(2*h^3);
p3 = 1/6 + (xip3-x)/(2*h) + (xip3-x).^2/(2*h^2) - (xip3-x).^3/(2*h^3);
p4 = (xip4-x).^3/(6*h^3);

c1 = (x >= xi).*(x < xip1);
c2 = (x >= xip1).*(x < xip2);
c3 = (x >= xip2).*(x < xip3);
c4 = (x >= xip3).*(x < xip4);

y = p1.*c1 + p2.*c2 + p3.*c3 + p4.*c4;


end
```

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> [ xplot, yplot1, yplot2, x, y, yplot ] = PlotCubicSpline( 0.1 );
>>
```



② **Linear multistep**

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j} + h \sum_{j=-1}^{p} b_j f_{n-j}, \quad n = p, p+1, \ldots$$

To find conditions when the scheme is consistent, consider

$$h\tau_{n+1}(h) = y_{n+1} - (a_0 y_n + a_1 y_{n-1} + \ldots + a_p y_{n-p}) - h(b_{-1} y'_{n+1} + b_0 y'_n + \ldots + b_p y'_{n-p})$$

$$h\tau_{n+1}(h) = y_n + y'_n h + O(h)^2 - a_0 y_n - a_1 (y_n - h y'_n + O(h^2)) - \ldots$$

$$- \ldots - a_p (y_n - p h y'_n + O(h^2))$$

$$- h\Big(b_{-1}(y'_n + O(h)) + b_0 (y'_n) + b_1 (y'_n + O(h)) + \ldots + b_p (y'_n + O(h)) \Big)$$

$$h\tau_{n+1}(h) = y_n (1 - a_0 - a_1 - \ldots - a_p)$$

$$+ h y'_n (a_1 + 2a_2 + \ldots + p a_p) - h(-1 + b_{-1} + b_0 + \ldots + b_p) y'_n + O(h^2)$$

So, $h\tau_{n+1}(h) = y_n \left(1 - \sum_{j=0}^{p} a_j\right) + h y'_n \left(\sum_{j=0}^{p} j a_j - \sum_{j=-1}^{p} b_j + 1 \right) + O(h^2)$

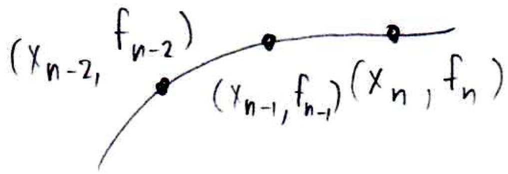
$$\Rightarrow \tau_{n+1}(h) = \frac{y_n}{h}\left(1 - \sum_{j=0}^{p} a_j\right) + y'_n \left(1 + \sum_{j=0}^{p} j a_j - \sum_{j=-1}^{p} b_j \right) + O(h).$$

Thus, the method is consistent if and only if

$$1 - \sum_{j=0}^{p} a_j = 0, \quad 1 + \sum_{j=0}^{p} j a_j - \sum_{j=-1}^{p} b_j = 0$$

$$\Longleftrightarrow \sum_{j=0}^{p} a_j = 1, \quad \text{and} \quad -\sum_{j=0}^{p} j a_j + \sum_{j=-1}^{p} b_j = 1.$$

③ Adams-Bashforth $\left(x = x_n,\ x_{n-1},\ x_{n-2} \right)$



$$P_2(x) = \gamma_0 + \gamma_1 (x - x_n) + \gamma_2 (x - x_n)(x - x_{n-1})$$

where

$$\gamma_0 = f_n$$

$$\gamma_1 = \frac{f_n - f_{n-1}}{h}$$

$$\gamma_2 = \frac{f_n - 2f_{n-1} + f_{n-2}}{2h^2}$$

$$\begin{array}{ll} x_n & f_n \\ x_{n-1} & f_{n-1} \\ x_{n-2} & f_{n-2} \end{array} \quad \frac{f_{n-1} - f_n}{x_{n-1} - x_n}, \quad \frac{f_{n-2} - f_{n-1}}{x_{n-2} - x_{n-1}} \quad \frac{\frac{f_{n-2} - f_{n-1}}{x_{n-2} - x_{n-1}} - \frac{f_{n-1} - f_n}{x_{n-1} - x_n}}{x_{n-2} - x_n} = \frac{\frac{f_{n-1} - f_{n-2}}{h} - \frac{f_n - f_{n-1}}{h}}{-2h}$$

$$P_2(x) = \gamma_0 + \gamma_1 (x - x_n) + \gamma_2 (x - x_n)\left[x - (x_n - h) \right]$$

$$= \gamma_0 + \gamma_1 (x - x_n) + \gamma_2 (x - x_n)^2 + h\gamma_2 (x - x_n)$$

$$\int_{x_n}^{x_{n+1}} P_2(x) = \gamma_0 h + \frac{\gamma_1}{2} (x - x_n)^2 \Big|_{x_n}^{x_{n+1}} + \frac{\gamma_2}{3} (x - x_n)^3 \Big|_{x_n}^{x_{n+1}} + \frac{h\gamma_2}{2} (x - x_n)^2 \Big|_{x_n}^{x_{n+1}}$$

$$= \gamma_0 h + \frac{\gamma_1}{2} h^2 + \frac{\gamma_2}{3} h^3 + \frac{h\gamma_2}{2} h^2$$

$$= f_n h + \frac{f_n - f_{n-1}}{2} h + \frac{f_n - 2f_{n-1} + f_{n-2}}{6} h + \frac{f_n - 2f_{n-1} + f_{n-2}}{4} h$$

$$= \frac{h}{12} \left[12 f_n + 6f_n - 6f_{n-1} + 2f_n - 4f_{n-1} + 2f_{n-2} + 3f_n - 6f_{n-1} + 3f_{n-2} \right]$$

$$= \frac{h}{12} \left[23 f_n - 16 f_{n-1} + 5 f_{n-2} \right]$$

So,

$$y_{n+1} = y_n + \frac{h}{12} \left[23 f_n - 16 f_{n-1} + 5 f_{n-2} \right]$$

From Taylor expansion: $\tau_{n+1} = \frac{3}{8} h^3 y_n^{(4)} + O(h^4)$

so, it has order 3.

Adams-Moulton

$(x = x_n, x_{n-1}, x_{n-2}, x_{n+1})$

$$P_3(x) = P_2(x) + \gamma_3 (x-x_n)(x-x_{n-1})(x-x_{n-2})$$

$$\gamma_3 = \frac{\dfrac{f_{n-1} - 2f_{n-2} + f_{n+1}}{2h^2} - \dfrac{f_n - 3f_{n-1} + 2f_{n-2}}{6h^2}}{h}$$

$$\gamma_3 = \frac{f_{n+1} - 3f_n + 3f_{n-1} - f_{n-2}}{6h^3}$$

Define

$$q(x) = \gamma_3 (x-x_n)(x-x_{n-1})(x-x_{n-2}) = \gamma_3 (x-x_n)\left[x-(x_n-h)\right]\left[x-(x_n-2h)\right]$$

$$= \gamma_3 (x-x_n)\left[(x-x_n)+h\right]\left[(x-x_n)+2h\right]$$

$$= \gamma_3 (x-x_n)^3 + 3h\gamma_3 (x-x_n)^2 + 2h^2\gamma_3 (x-x_n)$$

$$\int_{x_n}^{x_{n+1}} q(x)\,dx = \frac{\gamma_3}{4}(x-x_n)^4\Big|_{x_n}^{x_{n+1}} + h\gamma_3 (x-x_n)^3\Big|_{x_n}^{x_{n+1}} + h^2\gamma_3 (x-x_n)^2\Big|_{x_n}^{x_{n+1}}$$

$$= \frac{\gamma_3}{4}h^4 + h\gamma_3 h^3 + h^2\gamma_3 h^2$$

$$= \frac{9}{4}h^4\gamma_3 = \frac{3}{8}h\left(f_{n+1} - 3f_n + 3f_{n-1} - f_{n-2}\right)$$

So,
$$\int_{x_n}^{x_{n+1}} P_3(x)\,dx = \int_{x_n}^{x_{n+1}} P_2(x) + q(x)\,dx$$

$$= \frac{h}{12}\left[23f_n - 16f_{n-1} + 5f_{n-2}\right] + \frac{3h}{8}\left[f_{n+1} - 3f_n + 3f_{n-1} - f_{n-2}\right]$$

$$= \frac{h}{24}\left[46f_n - 32f_{n-1} + 10f_{n-2} + 9f_{n+1} - 27f_n + 27f_{n-1} - 9f_{n-2}\right]$$

$$= \frac{h}{24}\left[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}\right]$$

So,
$$y_{n+1} = y_n + \frac{h}{24}\left[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}\right]$$

From Taylor expansion, $\tau_{n+1} = -\frac{19}{720}h^4 y_n^{(5)} + O(h^5)$, so order = 4.

```
In[1]:= gam0 = fn;
        gam1 = (fn - fnm1) / h;
        gam2 = (fn - 2 * fnm1 + fnm2) / (2 * h^2);
        gam3 = (fnp1 - 3 * fn + 3 fnm1 - fnm2) / (6 * h^3);
        xnm1 = xn - h;
        xnm2 = xn - 2 * h;
        xnp1 = xn + h;
```

xnm1 = x_{n-1}
xnm2 = x_{n-2}
xnp1 = x_{n+1}

```
In[8]:= p2[x_] := gam0 + gam1 * (x - xn) + gam2 * (x - xn) * (x - xnm1);

In[9]:= p3[x_] := gam0 + gam1 * (x - xn) + gam2 * (x - xn) * (x - xnm1) + gam3 * (x - xn) * (x - xnm1) * (x - xnm2);

In[10]:= p3[xn]

Out[10]= fn

In[11]:= p3[xnm1]

Out[11]= fnm1

In[12]:= p3[xnm2]

Out[12]= 2 fn - 2 (fn - fnm1) - 2 fnm1 + fnm2

In[13]:= Simplify[2 fn - 2 (fn - fnm1) - 2 fnm1 + fnm2]

Out[13]= fnm2

In[14]:= p3[xnp1]

Out[14]= fnp1

In[15]:= Integrate[p3[x], {x, xn, xnp1}]
```

Out[15]= $\frac{19\,\text{fn}\,h}{24} - \frac{5\,\text{fnm1}\,h}{24} + \frac{\text{fnm2}\,h}{24} + \frac{3\,\text{fnp1}\,h}{8}$