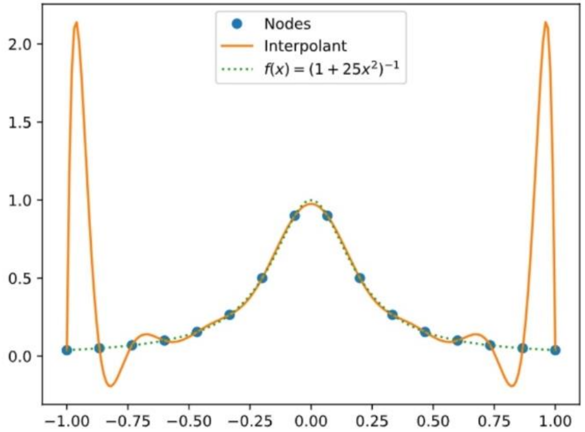
Runge- Phenomena

Important Topics on Curve-Fitting Techniques

Runge Phenomena in the interpolation of a smooth function

1. Introduction to Runge Phenomena

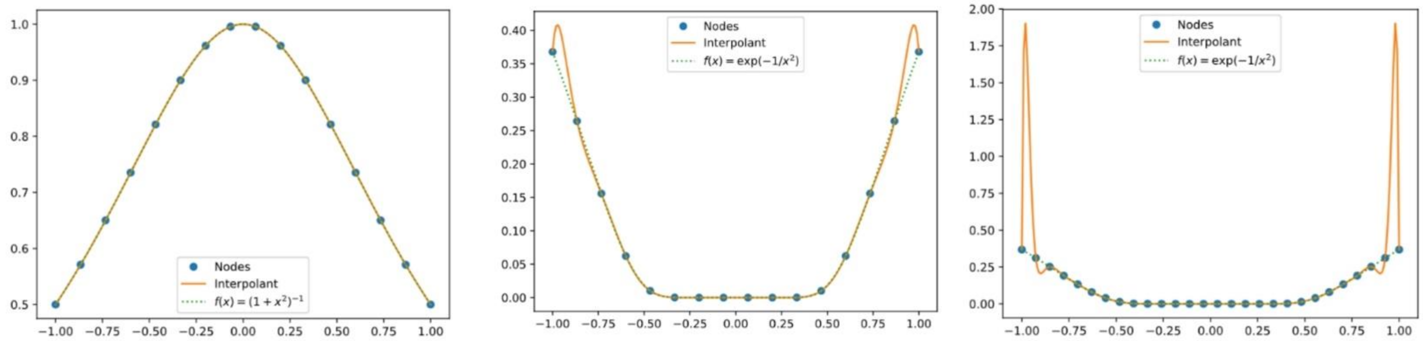
First of all, the "Runge" here is Carl David Tolmé Runge, better known for the Runge-Kutta algorithm for numerically solving differential equations. His name rhymes with cowabunga, not with sponge. Runge showed that polynomial interpolation at evenly-spaced points can fail spectacularly to converge. His example is the function $f(x) = 1/(1 + x^2)$ on the interval [-5, 5], or equivalently, and more convenient here, the function $f(x) = 1/(1 + 25x^2)$ on the interval [-1, 1]. Here's an example with 16 interpolation nodes.



Runge found that in order for interpolation at evenly spaced nodes in [-1, 1] to converge, the function being interpolated needs to be analytic inside a football-shaped region of the complex plane with major axis [-1, 1] on the real axis and minor axis approximately [-0.5255, 0.5255] on the imaginary axis. The function in Runge's example has a singularity at $0.2i$, which is inside the football. Linear interpolation at evenly spaced points would converge for the function $f(x) = 1/(1 + x^2)$ since the singularity at i is outside the football.

Curve-Fitting Techniques: Runge Phenomena

or another example, consider the function $f(x) = \exp(-1/x^2)$, defined to be 0 at 0. This function is infinitely differentiable but it is not analytic at the origin. With only 16 interpolation points as above, there's a small indication of trouble at the ends.



With 28 interpolation points in the plot below, the lack of convergence is clear.

Curve-Fitting Techniques: Runge Phenomena

2. Example of Runge Phenomena with Runge Function

(2-1) Runge Function

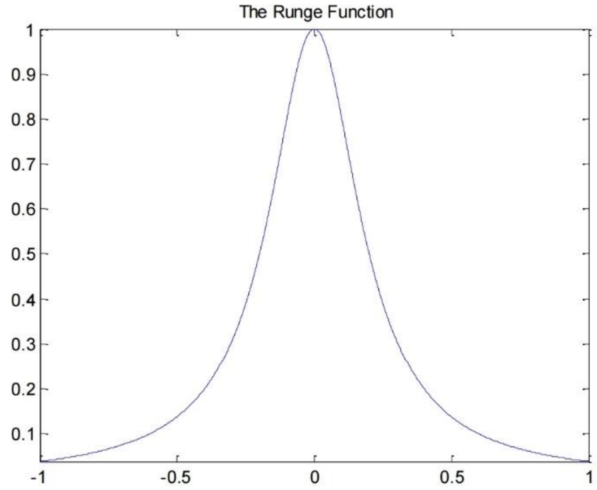
Definition of Runge Function

$$f(x) = \frac{1}{1+25x^2}$$

$$\frac{df(x)}{dx} = -\frac{50x}{\left(1+25x^2\right)^2}$$

Matlab program to plot

```
clear
clf
z = linspace(-1,1,1001);
f = @(x) 1 ./ (1 + 25.*x.^2);
plot(z,f(z))
title('The Runge Function')
axis tight
```



Curve-Fitting Techniques: Runge Phenomena

(2-2) Lagrange Interpolation of Runge Function to identify the Runge Phenomena

Program

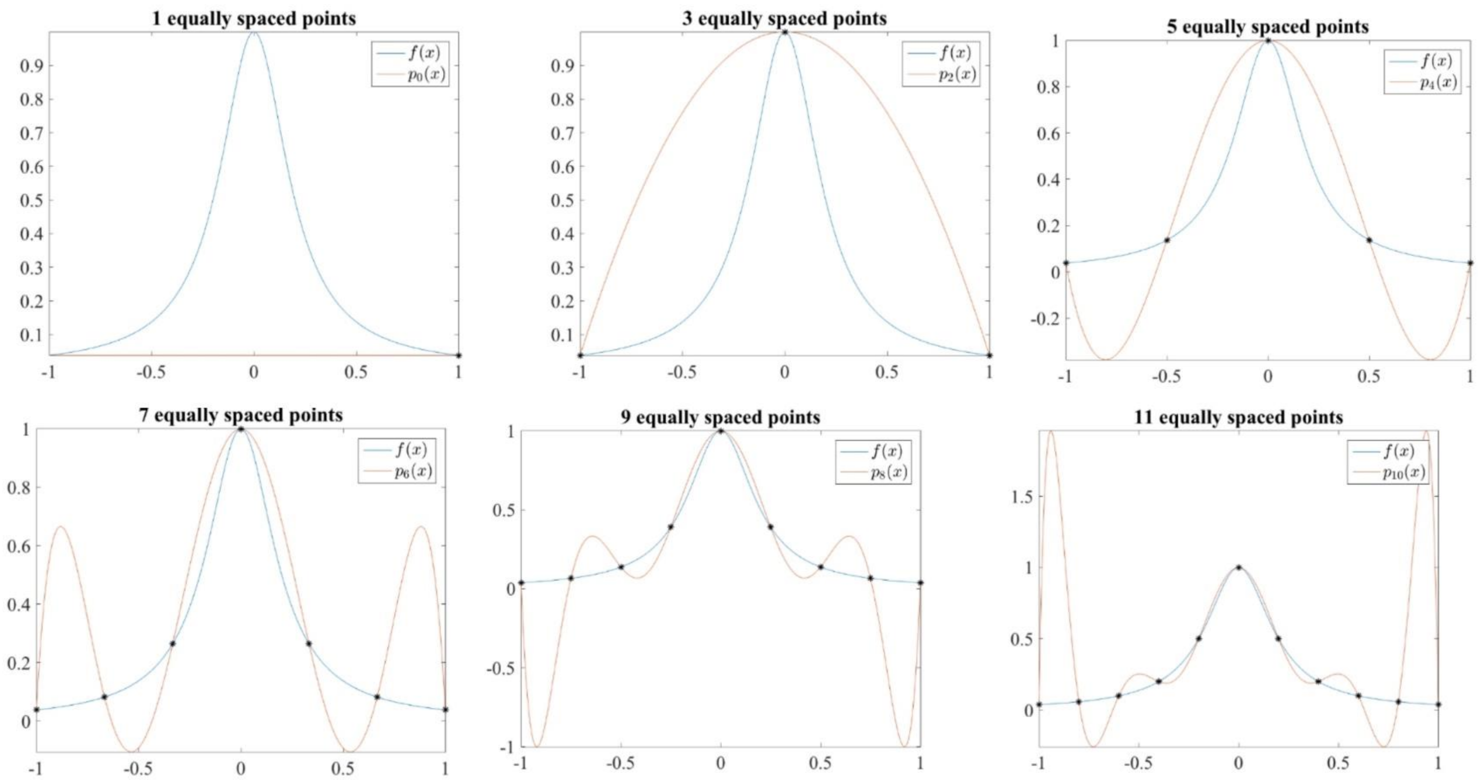
```
clear
clf
for N = 0:2:20
    x = linspace(-1,1,N+1);
    w = lagrange_weights(x);
    pn = lagrange_eval_naive(z, x, f(x), w);
    plot(z,f(z), z, pn, x, f(x), '*k')
    title(sprintf('%d equally spaced points', N+1))
    h = legend('$f(x)$', sprintf('$p_{%d}(x)$',N));
    set(h,'Interpreter','latex')
    axis tight
    xlim([-1 1])
    snapnow
end
```

```
% LAGRANGE_EVAL_NAIVE
% A naive implementation of the Lagrange interpolation function
%
% INPUTS:
% z     evaluation points [array of size m]
% x     abscissae [array of size n]
% y     function values at x(j) [array of size n]
% w     weights as computed using the Lagrange_weights functions [array of size n]
%
% OUTPUTS:
% f     interpolation to the points [array of size m]
   function Pn = Lagrange_eval_naive(z,x,y,w)

   m = length(x);
```

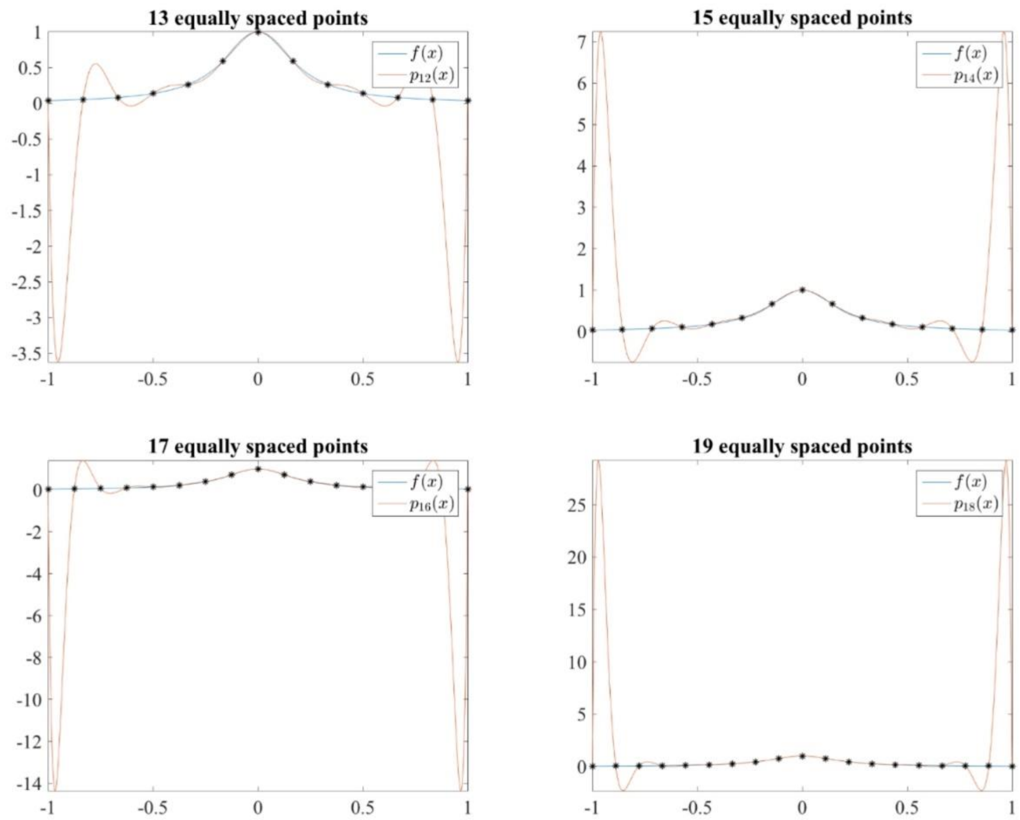
Curve-Fitting Techniques: Runge Phenomena

```
% compute the sum in the interpolation
Pn = 0;
for k = 1:m

    % computes the Lk for this point
    Lk = 1;
    % computes the given w(k)
    for j=1:m
        if j ~= k
            Lk = Lk .* (z-x(j));
        end
    end
    Lk = Lk * w(k);

    Pn = Pn + y(k)*Lk;
End

function w = Lagrange_weights(x)

m = length(x); % really m = n+1 in our formulas
for k = 1:m
    w(k) = 1;          % chooses the w(k) we are working on


    for j=1:m          % computes the given denominator of w(k)
        if j ~= k
            w(k) = w(k) * (x(k)-x(j));
        end
    end
    w(k) = 1/w(k);     % store the actual w(k)
end
```

Curve-Fitting Techniques: Runge Phenomena



Curve-Fitting Techniques: Runge Phenomena



Curve-Fitting Techniques: Runge Phenomena

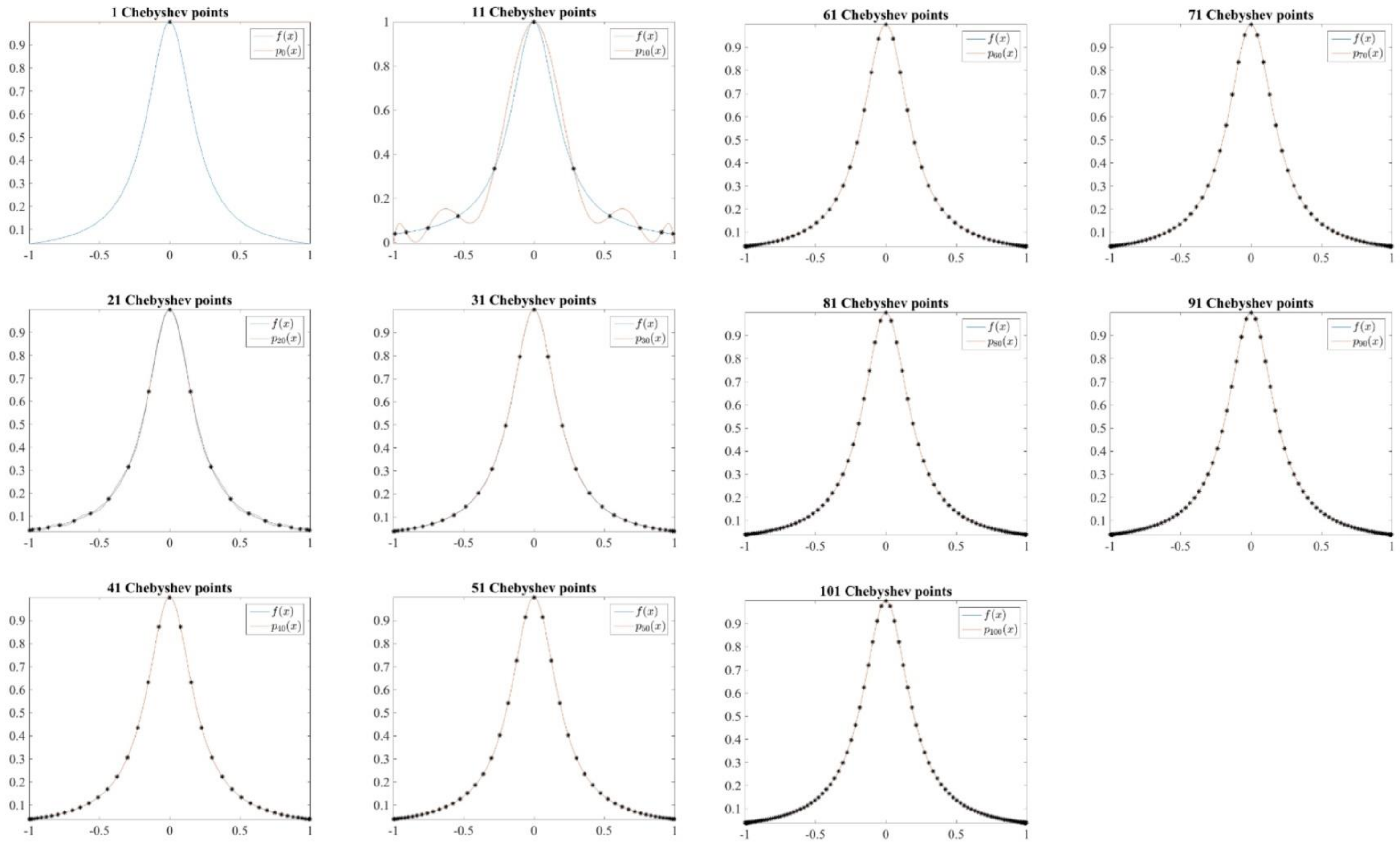
(2-3) Prevention of Runge Phenomena using Unequal-Spaced nodes

Example: Lagrange Interpolation Using Chebyshev nodes

Program

```
for N = 0:10:100
    x = chebyspace(-1,1,N+1);
    w = lagrange_weights(x);
    pn = lagrange_eval_naive(z, x, f(x), w);
    plot(z,f(z), z, pn, x, f(x), '*k')
    title(sprintf('%d Chebyshev points', N+1))
    h = legend('$f(x)$', sprintf('$p_{%d}(x)$',N));
    set(h,'Interpreter','latex')
    axis tight
    snapnow
end
```

Curve-Fitting Techniques: Runge Phenomena



(2-4) Hermite Spline Interpolation: Interpolation with the minimum Runge Phenomena

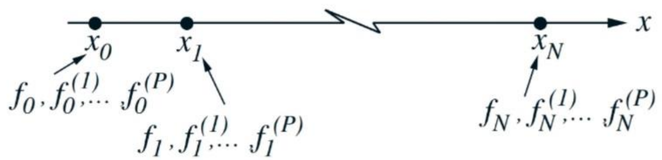
(2-4a) Reference

(Ref: A. SPITZBART, "A GENERALIZATION OF HERMITE'S INTERPOLATION FORMULA," University of Wisconsin-Milwaukee, The American Mathematical Monthly, Vol. 67, No. 1 (Jan., 1960), pp. 42-46)

(2-4b) Derivation in a Standard node $x \in [0,1]$

Hermite's interpolation formula provides an expression for a polynomial which passes through given points with given slopes.

For the given data set $\left\{\left(x_j, f_j, f_j^{(1)}, f_j^{(2)}, \cdots, f_j^{(P)}\right)\right\}_{j=0}^{j=n}$,the interpolating function can be expresses as



$$g(x) = \sum_{k=0} a_k x^k \quad \text{where} \quad f_j = f(x_j), \quad f_j^{(p)} = \frac{d^p f(x_j)}{dx^p}$$

The interpolating function should satisfy the following conditions

$$g(x_j) = f_j \quad (j = 0, \cdots, n)$$
$$g^{(1)}(x_j) = f_j^{(1)} \quad (j = 0, \cdots, n)$$
$$\vdots$$
$$g^{(p)}(x_j) = f_j^{(p)} \quad (j = 0, \cdots, n)$$

total $(p+1)(n+1)$ constraints with $g(x) = \sum_{k=0}^{k=(p+1)(n+1)-1} a_k x^k$.

Curve-Fitting Techniques: Runge Phenomena

[Example] Cubic Hermite interpolation with $p=n=1 \quad \rightarrow k=(p+1)(n+1)-1=3$

$$g(x)=a_0+a_1x+a_2x^2+a_3x^3 \qquad \text{with} \quad x\in[0,1]$$

$$\begin{pmatrix}1&0&0&0\\1&1&1&1\\0&1&0&0\\0&1&2&3\end{pmatrix}\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix}=\begin{pmatrix}f_0\\f_1\\f_0^{(1)}\\f_1^{(1)}\end{pmatrix}\rightarrow\begin{pmatrix}a_0\\a_1\\a_2\\a_3\end{pmatrix}=\begin{pmatrix}1&0&0&0\\0&0&1&0\\-3&3&-2&-1\\2&-2&1&1\end{pmatrix}\begin{pmatrix}f_0\\f_1\\f_0^{(1)}\\f_1^{(1)}\end{pmatrix}$$

Therefore, the interpolating polynomial can be represented by the Hermite basis functions.

$$\begin{aligned}g(x)&=f_0+f_0^{(1)}x+\left(-3f_0+3f_1-2f_0^{(1)}-f_1^{(1)}\right)x^2+\left(2f_0-2f_1+f_0^{(1)}+f_1^{(1)}\right)x^3\\&=\left(1-3x^2+2x^3\right)f_0+\left(3x^2-2x^3\right)f_1+\left(x-2x^2+x^3\right)f_0^{(1)}+\left(-x^2+x^3\right)f_1^{(1)}\\&=h_1^{00}(x)f_0+h_1^{01}(x)f_1+h_1^{10}(x)f_0^{(1)}+h_1^{11}(x)f_1^{(1)}\end{aligned}$$

$$h_1^{00}(x)=1-3x^2+2x^3$$

$$h_1^{01}(x)=3x^2-2x^3$$

$$h_1^{10}(x)=x-2x^2+x^3$$

$$h_1^{11}(x)=-x^2+x^3$$

Curve-Fitting Techniques: Runge Phenomena

(2-4c) Derivation in General node $x \in [x_0, x_f]$ **using nondimensional independent variable**

$$\tau = \frac{x - x_0}{x_f - x_0} \in [0,1] \qquad dx = (x_f - x_0)d\tau$$

$$\frac{df(x)}{dx} = \frac{df(x(\tau))}{d\tau}\frac{d\tau}{dx} = \frac{1}{(x_f - x_0)}\frac{dg(\tau)}{d\tau} = \frac{g'(\tau)}{(x_f - x_0)} \leftarrow g(\tau) = f\{x(\tau)\}$$

Interpolating function

$$g(\tau) = (1 - 3\tau^2 + 2\tau^3)g_0 + (3\tau^2 - 2\tau^3)g_f + (\tau - 2\tau^2 + \tau^3)g_0' + (-\tau^2 + \tau^3)g_f'$$

$$g'(\tau) = (-6\tau + 6\tau^2)g_0 + (6\tau - 6\tau^2)g_f + (1 - 4\tau + 3\tau^2)g_0' + (-2\tau + 3\tau^2)g_1'$$

$$\begin{array}{l} g(0) = g_0 \\ g(1) = g_f \\ g'(0) = g_0' \\ g'(1) = g_f' \end{array} \rightarrow \begin{array}{l} f(x_0) = g_0 \\ f(x_f) = g_f \\ \dfrac{df(x_0)}{dx} = \dfrac{g_0'}{x_f - x_0} \\ \dfrac{df(x_f)}{dx} = \dfrac{g_f'}{x_f - x_0} \end{array} \rightarrow \boxed{\begin{aligned} f(\tau) &= (1 - 3\tau^2 + 2\tau^3)f_0 + (3\tau^2 - 2\tau^3)f_f \\ &\quad + (x_f - x_0)\left\{\frac{df(x_0)}{dx}(\tau - 2\tau^2 + \tau^3) + \frac{df(x_f)}{dx}(-\tau^2 + \tau^3)\right\} \end{aligned}}$$

Curve-Fitting Techniques: Runge Phenomena

If we have three-point data of (x_0, f_0), (x_1, f_1), (x_f, f_f) with $x_0 < x_1 < x_f$,

$$\frac{df(x_f)}{dx}(-\tau^2+\tau^3) = \frac{f(\tau)-(1-3\tau^2+2\tau^3)f_0-(3\tau^2-2\tau^3)f_f}{(x_f-x_0)} - \frac{df(x_0)}{dx}(\tau-2\tau^2+\tau^3)$$

$$\frac{df(x_f)}{dx} = \frac{f(\tau)-(1-3\tau^2+2\tau^3)f_0-(3\tau^2-2\tau^3)f_f}{(x_f-x_0)(-\tau^2+\tau^3)} - \frac{df(x_0)}{dx}\frac{(\tau-2\tau^2+\tau^3)}{(-\tau^2+\tau^3)}$$

$$= \frac{f(\tau)-(1-3\tau^2+2\tau^3)f_0-(3\tau^2-2\tau^3)f_f}{(x_f-x_0)(-\tau^2+\tau^3)} + \frac{df(x_0)}{dx}\frac{(1-\tau)}{\tau}$$

Therefore, $$\boxed{\frac{df(x_f)}{dx} = \frac{f_1-(1-3\tau_1^2+2\tau_1^3)f_0-(3\tau_1^2-2\tau_1^3)f_f}{(x_f-x_0)(-\tau_1^2+\tau_1^3)} + \frac{df(x_0)}{dx}\frac{(1-\tau_1)}{\tau_1}}$$

The data at the mid-point can be used to estimate the final gradients. With the exact mid point of $\tau_1 = \dfrac{1}{2}$

$$\boxed{\begin{aligned}\frac{df(x_f)}{dx} &= \frac{-8(f_1-\frac{1}{2}f_0-\frac{1}{2}f_f)}{(x_f-x_0)} + \frac{df(x_0)}{dx}\\ &= \frac{df(x_0)}{dx} + \frac{4(f_0-2f_1+f_f)}{(x_f-x_0)} \leftarrow \frac{d^2f(x_1)}{dx^2} \approx \frac{(f_0-2f_1+f_f)}{\frac{1}{4}(x_f-x_0)^2}\\ &\approx \frac{df(x_0)}{dx} + \frac{d^2f(x_1)}{dx^2}(x_f-x_0)\end{aligned}}$$

Curve-Fitting Techniques: Runge Phenomena

The gradient at the mid-point can be computed

$$f(\tau) = (1-3\tau^2+2\tau^3)f_0 + (3\tau^2-2\tau^3)f_f + (x_f-x_0)\left\{\frac{df(x_0)}{dx}(\tau-2\tau^2+\tau^3) + \frac{df(x_f)}{dx}(-\tau^2+\tau^3)\right\}$$

$$\frac{df(\tau)}{d\tau} = (-6\tau+6\tau^2)f_0 + (6\tau-6\tau^2)f_f + (x_f-x_0)\left\{\frac{df(x_0)}{dx}(1-4\tau+3\tau^2) + \frac{df(x_f)}{dx}(-2\tau+3\tau^2)\right\}$$

$$\frac{df(t)}{dt} = \frac{1}{(x_f-x_0)}\frac{df(\tau)}{d\tau} = \frac{(-6\tau+6\tau^2)f_0 + (6\tau-6\tau^2)f_f}{(x_f-x_0)} + \frac{df(x_0)}{dx}(1-4\tau+3\tau^2) + \frac{df(x_f)}{dx}(-2\tau+3\tau^2)$$

$$\rightarrow \boxed{\frac{df(t_1)}{dt} = \frac{(-6\tau_1+6\tau_1^2)f_0 + (6\tau_1-6\tau_1^2)f_f}{(x_f-x_0)} + \frac{df(x_0)}{dx}(1-4\tau_1+3\tau_1^2) + \frac{df(x_f)}{dx}(-2\tau_1+3\tau_1^2)}$$

Curve-Fitting Techniques: Runge Phenomena

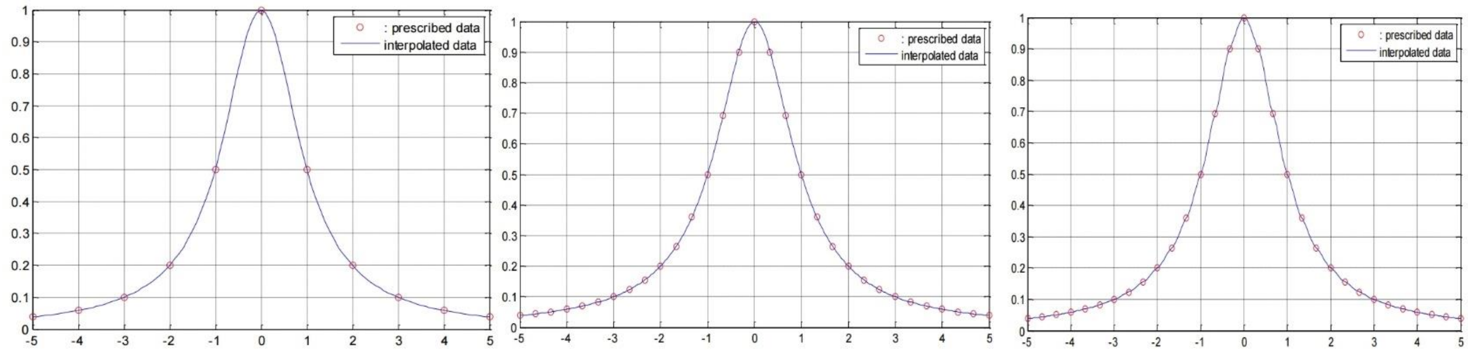
[Example#1] Application to the Modified Runge function

$$f(x)=\frac{1}{1+x^2}\quad,\qquad \frac{df(x)}{dx}=-\frac{2x}{\left(1+x^2\right)^2}$$

(a) With prescribed gradients at each node : node=11

(b) With gradients computed using Hermite algorithm at each node : node=31

(c) With gradients computed using Central difference formula at each node : node=31

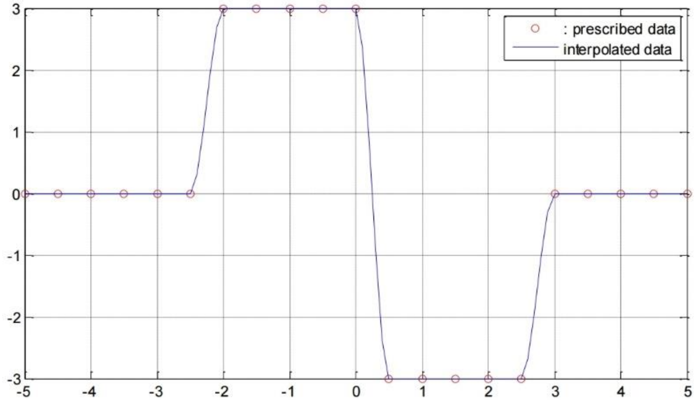


(a) (b) (c)

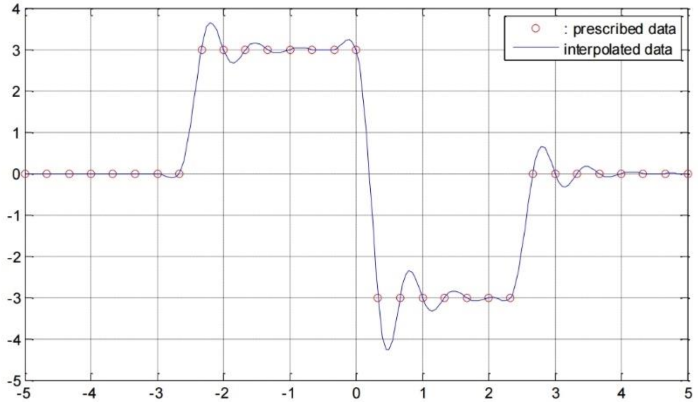
Curve-Fitting Techniques: Runge Phenomena

[Example#2] Application to the Doublet function

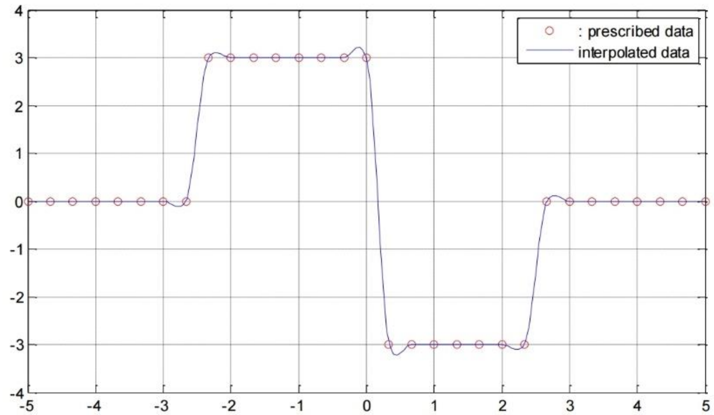
(a) With prescribed gradients at each node : node=21



(b) With gradients computed using Hermite algorithm at each node : node=31

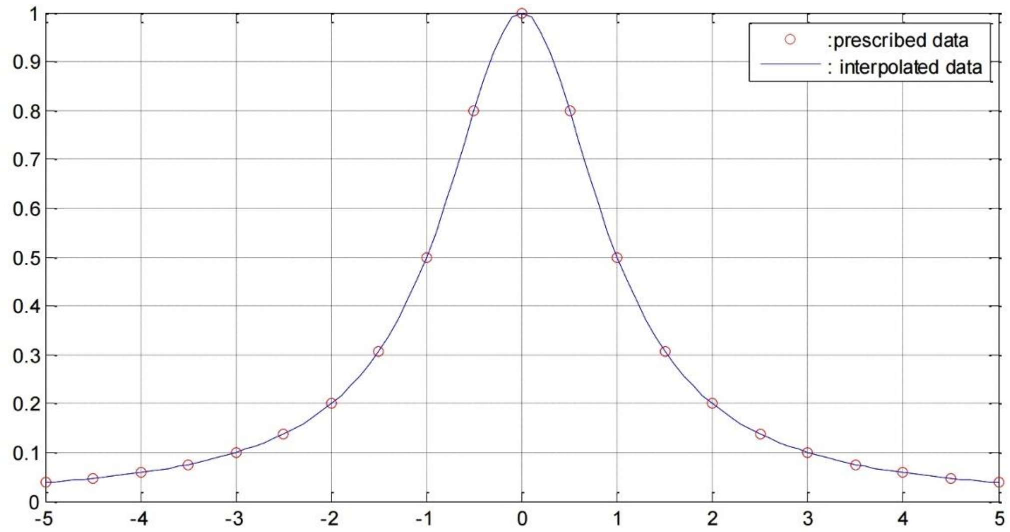


(c) With gradients computed using Central difference formula at each node : node=31



Curve-Fitting Techniques: Runge Phenomena

[Example #3] Test for modified Runge function : Quintic Hermit Polynomials



Curve-Fitting Techniques: Runge Phenomena

[Example #4] Comparison with difference interpolation methods

Piecewise Hermite Interpolation

Figure 2.5 shows some PCHIP interpolants to $f(x) = 1/(1+x^2)$ on the interval $[-5, 5]$. Compare with Figures 2.1 and 2.3.

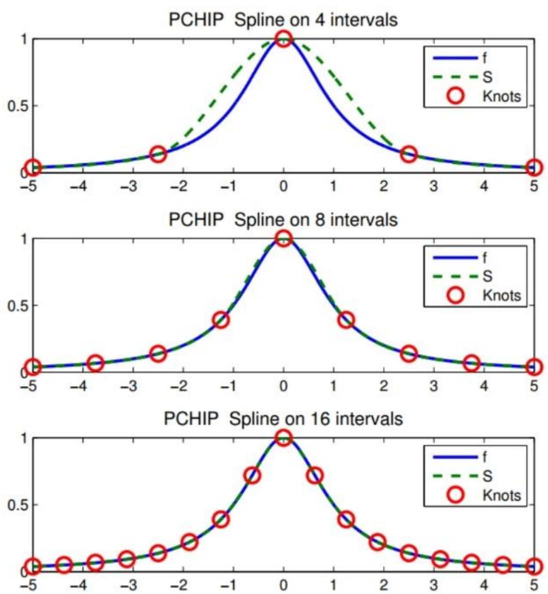
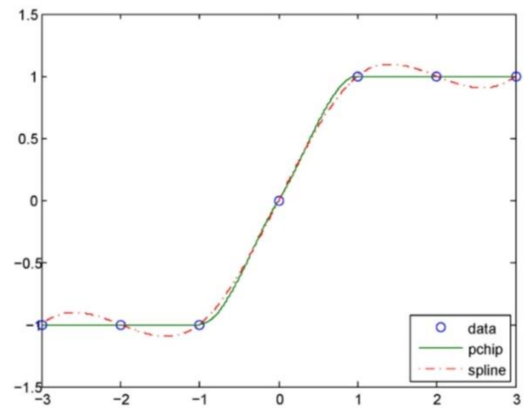
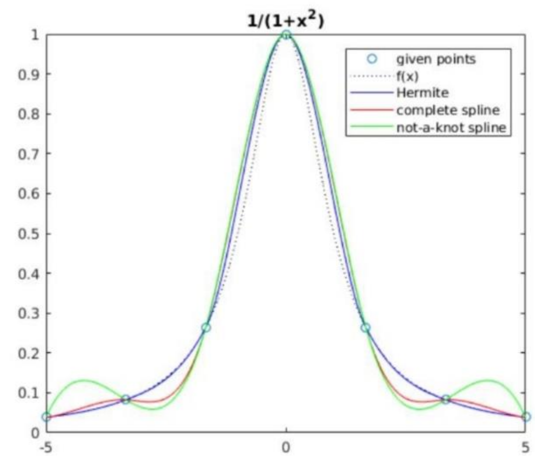


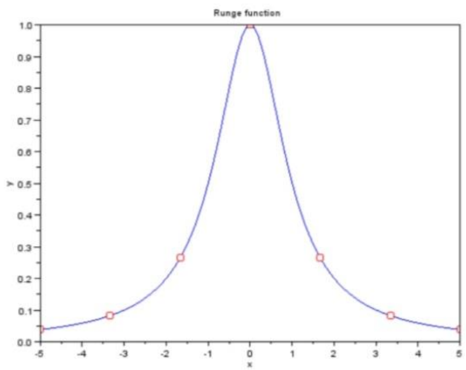
Fig. 2.5: PCHIP interpolants to $1/(1+x^2)$. Compare with Figure 2.3

Natural and Not-A-Knot Spline Interpolation

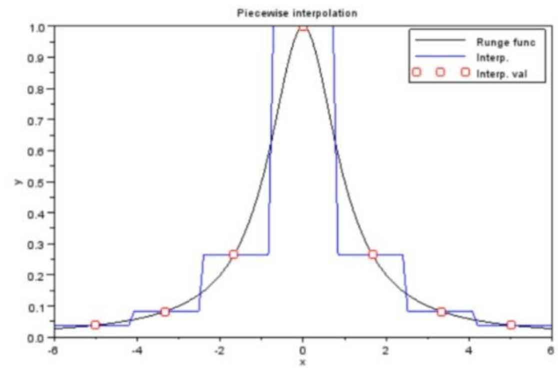




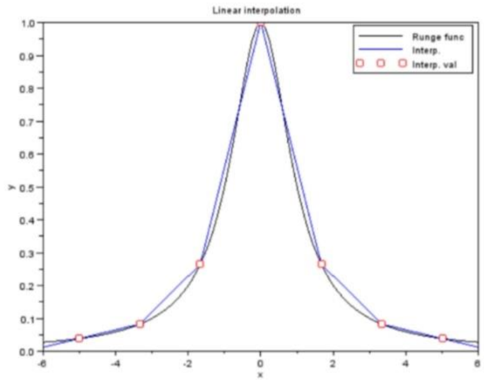
[Example #5] Polynomial Interpolation for Runge Function



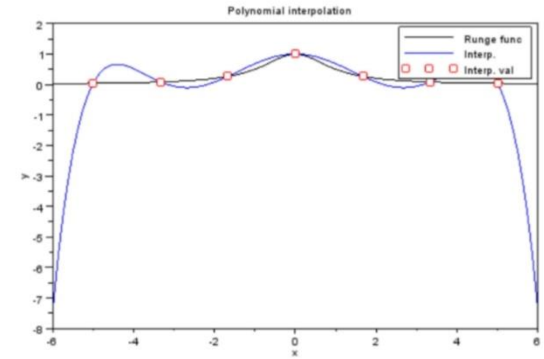
Interpolation function



Piecewise constant interpolation



Piecewise linear interpolation



Polynomial interpolation

End of Lecture