Examples of polynomial interpolation

• Go to MATLAB demo
  – Vandermonde
  – Polynomial interpolation for small set
  – For larger set
• See that even for six points we have a problem
  – In between the data points, (especially in first and last subintervals), function shows excessive variation.
  – overshoots changes in the data values.
  – As a result, full-degree polynomial interpolation is hardly ever used for data and curve fitting.
• However we saw polynomial interpolation works well when degree is low

Piecewise linear interpolation

• Simple idea
  – Connect straight lines between data points
  – Any intermediate value read off from straight line
• The local variable, \( s \), is
  \[ s = x - x_k \]
• The first divided difference is
  \[ \delta_k = (y_{k+1} - y_k)/(x_{k+1} - x_k) \]
• With these quantities in hand, the interpolant is
  \[ L(x) = y_k + (x - x_k) (y_{k+1} - y_k)/(x_{k+1} - x_k) \]
  \[ = y_k + s \delta_k \]
• Linear function that passes through \((x_k, y_k)\) and \((x_{k+1}, y_{k+1})\)
Piecewise linear interpolation

- Same format as all other interpolants
- Function diff finds difference of elements in a vector
- Find appropriate sub-interval
- Evaluate
- Jargon: $x$ is called a “knot” for the linear spline interpolant

```matlab
function v = piecelin(x,y,u)
% PIECELIN Piecewise linear interpolation.
% v = piecelin(x,y,u) finds piecewise linear L(x)
% with L(x(j)) = y(j) and returns v(k) = L(u(k)).
% First divided difference
delta = diff(y)./diff(x);
% Find subinterval indices k so that x(k) <= u < x(k+1)
n = length(x);
k = ones(size(u));
for j = 2:n-1
    k(x(j) <= u) = j;
end
% Evaluate interpolant
s = u - x(k);
v = y(k) + s.*delta(k);
```

So we can reduce error by choosing small intervals where 2nd derivative is higher
- If we can choose where to sample data
- Do more where the “action” is more

How good is piecewise linear interpolation?

Recall from Polynomial interpolation: If $f \in C^n[I]$, then

$$f(x) - p_{n-1}(x) = \frac{(x - x_1) \cdots (x - x_n)f^{(n)}(\xi)}{n!}$$

for some point $\xi$ in the interval containing $I$ and $x$.

We need to apply this to a polynomial of degree $n - 1 = 1$, so we obtain

$$f(x) - p_1(x) = \frac{(x - x_i)(x - x_{i+1})f''(\xi)}{2}$$

So we can reduce error by choosing small intervals where 2nd derivative is higher
- If we can choose where to sample data
- Do more where the “action” is more