Home Assignment 2 - Interpolation
Due 17.05.18

1. **Lagrange interpolation.** Implement the regular Lagrange interpolation algorithm, with the following signature:

   [yy]=LagrangeInterp(x, y, xx) ,

   where

   - **x** and **y** are the samples, i.e., $f(x_j) = y_j$. You can assume that the points are sorted and do not repeat.
   - **xx** and **yy** are the interpolated values, such that if $P_n(x)$ is the interpolation polynomial, then $P_n(xx_k) = yy_k$.

   Answer the following question: How do you verify that this is indeed Lagrange interpolation?

---

2. **Piecewise linear interpolation.** Build the piecewise linear approxiation. Same signature as **Lagrange**, but the function should be called **PWLinear**.

---

3. **Hermite interpolation.** Build the Hermite interpolant, by implementing the following function:

   [yy] = HermiteInterp(x, y, ytag, xx) ,

   where **x, y** and **xx** are as in the Lagrange interpolation, and **ytag** is the derivative at the sample points, i.e., $f'(x_i) = \text{ytag}_i$.

---

4. **Least-Squares Approximation.** Build the order n least-squares approximation by implementing

   [yy] = myLS(x, y, xx, n) ,

- **x** and **y** are the samples, i.e., $f(x_j) = y_j$.
- **xx** and **yy** are the approximated values, such that if $P_n(x)$ is the approximating polynomial, then $P_n(xx_k) = yy_k$.
- **n** is the degree of the polynomial approximation.

---

5. **Compare methods**. For the following functions, use N samples on the interval $[-1, 1]$, with $N = 2, 4, 8, \ldots 256$. Use a uniform grid of size N. For each method and for each function from the list below, compute the interpolant/approximation on **xx= linspace(-1,1,1e4)** once when the interpolation is with clean samples **y=f(x)** and the second time with added random noise **y2 = f(x) + 0.01*rand(size(x))**. For each function, plot maximal-pointwise error vs. the grid size N *for all four methods on the same figure*. Use **loglog** or **semilogy** to plot. All in all, you should end up with a single figure for each of these functions:

   (a) $f(x) = x$

   (b) $f(x) = x^8 + 6x^3$

   (c) $f(x) = \tanh(9x) + \frac{x}{2}$

   (d) $f(x) = \sin(20x)(1 + x)$

   (e) $f(x) = |x|$

   (f) $f(x) = \frac{1}{1+16x^2}$

---

6. **ImageRestoration** Import the image **gate.jpg** by double clicking on it when in Matlab. plot it by **imagesc(gate)**.

   (a) Reduce it to a low-quality version by **badGate = gate(1:10:end,:,:)**. Plot it as well.

   (b) Interpolate it back to the original resolution using both piecewise-linear interpolation and Lagrange interpoaltion. Plot these as well.

   (c) Repeat the same exercise with **badGate = gate(1:3:end,:,:)**.

   (d) In both cases, polynomial interpolation looks awful. What can you do to fix it, while still using polynomial interpolation? Provide code and ”fixed” image.

   Technical notes:

   - The imported image will be in unsigned int format, i.e., **uint8**. To perform analysis on it, you need to cast it to **double**. To plot back the results, cast it again to **uint8**.
   - To plot all 4 images on the same figure, use the **subplot** command.
   - To give a title to each subplot, use the **title** command.

