Matlab Homework 4a

In the online book:

- Do the “Challenge Activities” of: 3.5-7; 6.14,15
- Do the “Participation Activities” of: 5.6

Matlab Homework 4b

The same general requirements as for homework 3b apply. And you must study the posted lesson(s) and have done the online book part above before you can ask a TA or the instructor for help.

1. You measured a function f that, “unknown to you,” is exactly equal to

$$f_{\text{exact}}(t) = \frac{1}{2} + \sin\left(\frac{t}{2}\right) + \frac{1}{3}e^{t/3}$$

Create a handle `fExactFun` to an anonymous function that evaluates f at a given time t as above. Check that at time $t = 1$, your function gives you 1.44 as it should.

Now assume you did 8 measurements at equally spaced times from -3 to 3. Put the measured times in array `tMeasured`. Determine what the measured f-values would have been and put them in array `fMeasured`.

Evaluate the linear and spline interpolated f-values at times -3.5, -1.5, 0, 1.5, and 3.5. (Note: Do each of these times separately, one after the other). Print the obtained values and their errors out in the format

```
At t =  1.1:
the linear interpolate is  1.123, with error 1.12E12;
the spline interpolate is  1.123, with error 1.12E12;
the error in the spline is 12.1 times smaller.
```

using a `fprintf` for each line.

There may absolutely not be any printed numbers in the `fprintf` *format strings.* Only %...`f` and %...`e` number formatting operators may be in the format string. The actual numbers must be taken from variables `t`, `fLinear`, `fSpline`, `errLinear`, and `errSpline`. The numbers must line up as shown. The number formats must allow for the negative sign of some of the times and interpolates. (To get the errors, take the absolute value of the difference between interpolated and exact at that time.)

For the linear interpolation, do not forget that -3.5 and 3.5 are extrapolations, not interpolations. Modify the call appropriately.

2. Create a plot of the interpolations. Plot the exact solution as a black dashed line. Use 200 plot points from -4.5 to 4.5 to plot it. Also plot the measured data as black circles in the plot. And also plot the linear and spline interpolates at the 200 plot points in the plot. Use red for the linear interpolate, blue for the spline one.

   Your horizontal axis should go from -4.5 to 4.5 and your vertical axis from -1 to 3. Use title "Comparison of Interpolations", and axis labels "t" and "f". Include curve labels "Exact", "Measured", "Linear" and "Spline" to the graph. (See the posted lesson on how you can add curve labels using the `legend` command. Use `help legend` to figure out how to put the legend in a better position in the graph than on top of the curve.)

3. Next, compute and print out, using suitable `fprintf` commands,

   - The maximum error in the linear interpolation for the plot points (which includes extrapolation).
   - The maximum error in the spline interpolation for the plot points (which includes extrapolation).
   - The maximum error in the linear interpolation in the interpolation range (i.e. from -3 to 3). (For this and the next item, use about 100 equally spaced points in the interpolation range.)
   - The maximum error in the spline interpolation in the interpolation range (i.e. from -3 to 3).

   Comment on the results using `disp`.

4. Next recompute the measured f-values, but assume that they now have random errors of about 0.1 in average magnitude. In particular, Matlab users create the new measured data as follows:

```
rng('default')
fMeasured=fExactFun(tMeasured)+0.1*randn(size(tMeasured));
```

   Octave creates different random numbers than Matlab, so Octave users should instead use

```
fMeasured=fExactFun(tMeasured)+0.1*...
  [0.54 1.83 -2.26 0.86 0.32 -1.31 -0.43 0.34];
```

   to ensure that they get comparable results.

   Now repeat the first three questions and comment on the results where appropriate.