Lab exercises on polynomial interpolation

Course on Mechanical Engineering, AY 2015-16

Prof. S. De Marchi

Padova, May 9, 2016

1 Divided differences and Hörner scheme

We start by provinding two codes necessary for today's exercises.

```
function [d]=DiffDivise(x,y)
%----------------------------------------------------------------
% This function implements the algorithm of divided differences
%----------------------------------------------------------------
% Inputs
% x: vector of interpolation points,
% y: vector of function values.
%
% Output
% d: vector of divided differences
%----------------------------------------------------
n=length(x); d=y;
for i=2:n,
    for j=2:i,
       d(i)=(d(i)-d(j-1))/(x(i)-x(j-1));
    end;
end;

function [p]=Horner(d,x,xe)
%-------------------------------------------------------------------------
% This function implements the Horner scheme for evaluating
% the interpolating polynomial in Newton form on a set of evaluation points
%-------------------------------------------------------------------------
% Inputs
% d: divided differences vector
% x: vector of interpolating points
% xe: vector of evaluation points
%
% Output
% p: the polynomial evaluated at all targets
%----------------------------------------------
n=length(d);
for i=1:length(xe);
 p(i)=d(n);
 for k=n-1:-1:1
  p(i)=p(i)*(xe(i)-x(k))+d(k);
 end
end
```

Solve the following exercises in Matlab

1. Construct the interpolating polynomial in Newton form, of degrees $n = 5, \ldots, 10$ of the *scaled Runge function*
$$g(x) = \frac{1}{1 + 25\,\epsilon^2\,x^2}, \quad x \in [-1, 1]$$
on Chebyshev points. Consider two values of *shape parameter* $\epsilon = 0.2, 6$.

   Write a M-function that takes as input n and ϵ and determines the interpolating polynomial and returns the corresponding relative errors (varying n and ϵ).

2. Consider the function $f(x) = x + \mathrm{e}^x + \dfrac{20}{1+x^2} - 5$ restricted to the interval $[-2, 2]$. Write an M-function that

   (a) determines the interpolating polynomial of degree 5 in *Newton form* on the equispaced points $x_k = -2 + kh,\ k = 0, \ldots, 5$.

   (b) Compute an overestimate of the interpolation error in a chosen point $x^* \in (-2, 2)$.

   Repeat the calculations by using Chebyshev points.

3. Consider the function $f(x) = \sin(x) + \sin(5x),\ \ x \in [0, 2\pi]$. Write a script that

   $\boxed{\text{i}}$ Constructs the interpolating cubic spline (use `s=spline(x,y,xx)` where `x=0:h:`2π is a vector of equispaced nodes (with step $h = 2\pi/n$), `y` is the vector of the values of f at the nodes and `xx` is a vector of target points). Take $n = 5$.
   `PS`: notice that taking $n = 6$ we sample the spline at the zeros of the function and the corresponding cubic spline is equal to 0.

   $\boxed{\text{ii}}$ Construct also the approximating polynomial of degree $n = 6$ on the same set of equispaced points `x=0:h:`2π computed by the command `polyfit` and evalutated at the targets using `polyval`.

   $\boxed{\text{iii}}$ In both cases, compute the 2-norm of the error. Among (i) and (ii) which one approximates $f(x)$ better?

   $\boxed{\text{iv}}$ Repeat the previous steps with a smaller $h = h/2,\ h/4,\ h/8,\ h/16$, that implies to double n at each new step size.

4. On the interval $[-1, 1]$ compute the *Lebesgue constant* on a set of $n = 1, \ldots, 100$ equispaced points and a set of $n = 1, \ldots, 100$ Chebyshev points. *Hints:* for the evaluation of all Lagrange polynomials, take a fine set of $m = 10000$ (or more) evalution points where construct the matrix L, $m \times n$, whose columns are the n Lagrange polynomials at the m evaluation points. The Lebegsue constant is then $\|L\|_\infty$, i.e. `norm(L,inf)`.

5. Take the function $f(x) = \dfrac{20}{1 + \log(x^2)} - 5\,\sin(\mathrm{e}^x)$ restricted to the interval $[1, 2]$. Find the unique interpolating polynomial of degree 2, $p_2(x) = a_0 + a_1 x + a_2 x^2$ such that
$$p_2(1) = f(1),\ p_2(2) = f(2),\ \int_1^2 p_2(x)dx = \int_1^2 f(x)dx\ .$$

Find it by solving the 3×3 linear system.

To compute the integral, use the built-in function `quadl` of Matlab

(use it as `integral=quadl(fun,a,b,tol)`)

with tolerance $tol = 1.e - 6$. Then make the graphs of the function, of the polynomial and of the error $\|f - p_2\|_\infty$.

Solve the same problem by using three equispaced points $0, 1/2, 1$ (i.e. by solving the corresponding Vandermonde system).

Time: **2 hours**.