Computational methods and applications (AMS 147)

Homework 3, due Wednesday Feb 14

Please submit to CANVAS a .zip file that includes the following Matlab functions:

`Lagrange_interp.m`
`test_Lagrange_interpolation.m`
`compute_Lebesgue_function.m`
`test_Lebesgue_function.m`

Exercise 1 Write a function `Lagrange_interp.m` that computes the Lagrangian interpolant of a given set of data points (x_i, y_i), $i = 1, 2, \ldots$. The function should be of the form

```
function [y] = Lagrange_interp(xi,yi,x)
```

Input:
`xi`: vector of interpolation nodes
`yi`: vector of data points at interpolation nodes
`x`: vector of points at which we evaluate the polynomial interpolant

Output:
`y`: polynomial interpolant evaluated at `x`

<u>*Hint*</u>: Compare the output of your function with the output of the Matlab/Octave built-in function,

```
y=polyval(polyfit(xi,yi,length(xi)-1),x)
```

(see the Matlab/Octave documentation).

Exercise 2 Consider the nonlinear function

$$f(x) = \frac{1}{1 + 20x^2}, \qquad x \in [-1, 1]. \tag{1}$$

By using the Matlab function you coded in Exercise 1, determine the Lagrangian interpolant of f, i.e. the polynomial $\Pi_N f(x)$ that interpolates the set of data $\{x_i, f(x_i)\}_{i=0,..,N}$ in the following cases:

1. Evenly-spaced grid with $N+1$ points

$$x_j = -1 + 2\frac{j}{N}, \qquad j = 0, .., N \tag{2}$$

2. Unevenly-spaced grid with $N+1$ points (Chebyshev-Gauss-Lobatto points)

$$x_j = \cos\left(\frac{\pi}{N} j\right), \quad j = 0, 1, ..., N, \tag{3}$$

In particular, write a Matlab function `test_Lagrange_interpolation.m`

```
function [x,f,P1,P2,P3,P4]=test_Lagrange_interpolation()
```

that returns the follwing items:
`x`: vector of 1000 evenly-spaced nodes in $[-1,1]$ (use the command `x=linspace(-1,1,1000)`).
`f`: vector representing (1) evaluated at `x`.
`P1`: Lagrangian interpolant of (1) built on the grid (2) with $N = 8$ nodes and evaluated at `x`.
`P2`: Lagrangian interpolant of (1) built on the grid (2) with $N = 20$ nodes and evaluated at `x`.
`P1`: Lagrangian interpolant of (1) built on the grid (3) with $N = 8$ nodes and evaluated at `x`.
`P1`: Lagrangian interpolant of (1) built on the grid (3) with $N = 20$ nodes and evaluated at `x`.

The function should also plot (1) (in blue) and the Lagrangian interpolants (in red) obtained by using both the evenly-spaced and the unevenly-spaced grids for the cases $N = 8$ and $N = 20$ (4 different figures). Each figure should include the graph of $f(x)$, the data points $\{x_i, f(x_i)\}$ and the interpolant $\Pi_N f(x)$ through those points.

Hint: See the code uploaded in CANVAS for examples of similar plots.

Exercise 3 Let $\{l_i(x)\}_{i=0,..,N}$ be the set of Lagrange characteristic polynomials associated with the nodes $\{x_j\}_{j=0,..,N}$. We have seen in class that the polynomial interpolation error is related to the Lebesgue function

$$\lambda_N(x) = \sum_{j=0}^{N} |l_j(x)| \qquad \text{(Lebesgue function)}, \tag{4}$$

and the Lebesgue constant

$$\Lambda_N = \max_{x\in[-1,1]} \lambda_N(x) \qquad \text{(Lebesgue constant)}. \tag{5}$$

1. Given the vector of interpolation nodes `xi=[xi(1) ...  xi(N+1)]`, write a Matlab/Octave function `compute_Lebesgue_function.m` that returns the Lebesgue function (4) evaluated at

1000 evenly-spaced nodes between `xi(1)` and `xi(N+1)`. Such function should also return the Lebesgue constant (5).

```
function [lambda,L]=compute_Lebesgue_function(xi)
```

Input:

`xi`: vector of interpolation nodes `xi=[xi(1) ... xi(N+1)]`

Output:

`lambda`: Lebesgue function $\lambda_N(x)$ evaluated at 1000 evenly-spaced nodes between `xi(1)` and `xi(N+1)`.

`L`: Lebesgue constant Λ_N.

2. Apply the function `compute_Lebesgue_function(xi)` to the four cases of evenly- and unevenly-spaced grids you studied in Exercise 2. To this end, write a function

```
function [L1,L2,L3,L4]=test_Lebesgue_function()
```

that plots the Lebesgue function (4) corresponding to the aforementioned four cases (in 4 different Figures), and returns the value of the Lebesgue constant for each case.

Remark: Recall, that the smaller the Lebesgue constant the smaller the approximation error of the Lagrangian polynomial interpolation. If fact, we have seen in class that

$$\|f(x) - \Pi_N(x)\|_\infty \leq (1 + \Lambda_N) \inf_{\psi \in \mathbb{P}_N} \|f(x) - \psi(x)\|_\infty \tag{6}$$