INTERNATIONAL UNIVERSITY OF SARAJEVO

SPRING 2017.

MATH 205 NUMERICAL ANALYSIS
FINAL EXAM

STUDENT NAME & ID: ____________________________

DATE: ____________________________

Instructions:

• Examination time: **120 min**.

• Write your **name** and **student ID number** in the space provided above.

• This examination is **closed** book and **closed** notes. You may **use a calculator** to perform the calculations required.

• There are 3 questions. The points for each question are given in brackets, next to the question title. The overall maximum score is 100. **This exam weighs 40% of your final grade.**

• Answer each question in the space provided. If you need to continue an answer onto the back of the sheet, clearly indicate that and label the continuation with the question number.

QUESTION	**1**	**2**	**3**
POINTS	**/20**	**/40**	**/40**
		TOTAL	**/100**

1. Answer and justify (20 %)

1.1 Name a few important criteria in assessing methods for finding roots! **(5)**

1.2 You need to calculate the definite integral of the polynomial $f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

$$I = \int_0^2 f(x)dx$$

(7)

Explain which of the following may be more accurate? The analytical solution

$$I = \left({a_3}/{4}\, x^4 + {a_2}/{3}\, x^3 + {a_1}/{2}\, x^2 + a_0 x \right)_0^2 = 4a_3 + {8}/{3}\, a_2 + 2a_1 + 2a_0$$

or the Simpsons integration

$$I = \frac{2}{6}\left(f(0) + 4f(1) + f(2) \right)$$

1.3 What is full polynomial and what is piece-wise polynomial interpolation? What is spline? Why is it sometimes good idea to use piece-wise interpolation instead of full polynomial interpolation? **(8)**

2. General (40%)

2.1 Let x be the smallest **positive** real solution of $tan(x) = -x$, with ***x* in radians**. Use the **bisection method** to find x accurate to **two significant digits**. Use the interval $1.8 < x < 2.3$. How many iterations of the bisection method would be required to guarantee that the **absolute** error in the solution is less than 10^{-6}? **(10)**

Number of iteration	x_l	x_u	Max error
0	**1.8**	**2.3**	
1			
2			
3			
4			
5			
6			

2.2 Using Newton-Raphson's method, write the first three iterations for the equation below! Use $x_0 = 1$. Why is it not smart to have $x_0 = 1/\sqrt{3}$ as the initial guess? **(10)**

$$x^3 - x = 1$$

Number of iteration	Guess	Error
0	**1**	**---**
1		
2		
3		

2.3 Let $\operatorname{sinc}(t) = \begin{cases} 1 & \text{if } t = 0 \\ \frac{\sin t}{t} & \text{if } t \neq 0. \end{cases}$ The function $f(x) = \int_0^x \operatorname{sinc}(t)\, dt$ **(10)**

Use the **composite** Simpson's rule with n = 4 (five points) to evaluate $f(\pi)$.

3. Matlab (40%)

3.1 You have a dataset of 7 measurements of solar power collected by a panel, sampled every hour at discrete times t_n with the following values $f(t_n)$:

t [h]	0	1	2	3	4	5	6
f(t) [W]	0.1201	4.7509	8.4490	10.3328	9.3699	5.8497	1.7147

```
t=[0 1 2 3 4 5 6];
f=[0.1201 4.7509 8.4490 10.3328 9.3699 5.8497 1.7147];
```

a) Write a Matlab program that interpolates a full polynomial to the dataset! **(5)**

b) Write a Matlab program that approximates the data by a 5^{th} degree polynomial! **(5)**

c) Write a Matlab program that interpolates the data by a cubic spline! How do you evaluate, using the spline, the function at `t=3.5` **(5)**

BONUS: Find the time at which solar irradiation is at its maximum, using the spline interpolation

d) It turns out that approximation of the data by trigonometric functions gives best results. The least squares approximation says the function is **(5)**

```
fun=@(tt) 5.7982+1.5216*sin(0.8976*tt)-4.7948*cos(0.8976*tt);
```

How can you approximate the total energy collected by the panel that day? (How do you integrate the function `f(t)` … in Matlab?)

e) How can you estimate after what time was the energy collected 20Wh? You may use the `fzero` command to find this! **(5)**

f) What is the rate of change of solar irradiation 2 hours after the experiment started? Use forward-, backward-, central- **and/or** Richardson formula to get the derivative! **(5)**

3.2 A system of coupled ordinary differential equations is given below. The initial conditions are $y_1(0) = 0$, $y_2(0) = -1$. You are interested in values (and behavior) of $y_1(0)$ to $y_1(5)$. How can you solve the system and plot the solution using Matlab? **(10)**

$$\frac{dy_1}{dt} = -4y_1 - 2y_2 + \cos(t)$$

$$\frac{dy_2}{dt} = 3y_1 + y_2$$

Enjoy the summer!

You may find useful some of the following formulae and Matlab commands:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad x_{i+1} = x_i - f(x_i)\frac{x_{i-1} - x_i}{f(x_{i-1}) - f(x_i)} \qquad |x^* - \texttt{root}| \le \frac{b-a}{2^n}$$

$$I = (b-a)\frac{f(x_0) + 4\sum_{i=1,3,5}^{n-1} f(x_i) + 2\sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)}{3n} \qquad f(x) = \sum_{k=1}^{N}\left(\prod_{\substack{j=1\\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}\right) y_k$$

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} \qquad f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} \qquad f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

```
>> help polyfit
 POLYFIT Fit polynomial to data.
    P = POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X) of
    degree N that fits the data Y best in a least-squares sense. P is a
    row vector of length N+1 containing the polynomial coefficients in
    descending powers, P(1)*X^N + P(2)*X^(N-1) +...+ P(N)*X + P(N+1).
```

```
>> help spline
 SPLINE Cubic spline data interpolation.
    PP = SPLINE(X,Y) provides the piecewise polynomial form of the
    cubic spline interpolant to the data values Y at the data sites X,
    for use with the evaluator PPVAL and the spline utility UNMKPP.
    X must be a vector.
```

```
>> help quad
 QUAD   Numerically evaluate integral, adaptive Simpson quadrature.
    Q = QUAD(FUN,A,B) tries to approximate the integral of scalar-valued
    function FUN from A to B to within an error of 1.e-6 using recursive
    adaptive Simpson quadrature. FUN is a function handle. The function
    Y=FUN(X) should accept a vector argument X and return a vector result
    Y, the integrand evaluated at each element of X.
```

```
>> help fminbnd
 FMINBND Single-variable bounded nonlinear function minimization.
    X = FMINBND(FUN,x1,x2) attempts to find  a local minimizer X of the function
    FUN in the interval x1 < X < x2.  FUN is a function handle.  FUN accepts
    scalar input X and returns a scalar function value F evaluated at X.
```

```
>> help ppval
 ppval  Evaluate piecewise polynomial.
    V = ppval(PP,XX) returns the value, at the entries of XX, of the
    piecewise polynomial f contained in PP, as constructed by PCHIP, SPLINE,
    INTERP1, or the spline utility MKPP.
```

```
>> help polyval
 polyval Evaluate polynomial.
    Y = polyval(P,X) returns the value of a polynomial P evaluated at X. P
    is a vector of length N+1 whose elements are the coefficients of the
    polynomial in descending powers.
```