به نام خدا

دانشکده فنی دانشگاه تهران

محاسبات عددی

استاد: دکتر آریانیان

پروژه 5

محمدمهدی عبدالحسینی

810198434

نیمسال اول 00-99

بخش اول:

با دانشی که از درس ریاضی 1 بدست آوردیم، برای یافتن حاصل حد میتوان بصورت زیر عمل کرد:

$$\lim_{x\to 0}\frac{1-cos(x)}{x^2}=\frac{0}{0} \Rightarrow HOP \Rightarrow \lim_{x\to 0}\frac{sin(x)}{2x} = \lim_{x\to 0}\frac{x-\frac{x^3}{3!}+\frac{x^5}{5!}-\cdots}{2x}$$
$$\Rightarrow \lim_{x\to 0}\frac{1-\frac{x^2}{3!}+\frac{x^4}{5!}+\cdots}{2}=\frac{1}{2}$$

```
f = @(x) (1-cos(x))/x^2;
a = pi*10 ^ (-8);
f(a)
```

برای تعیین مفهوم خط اول، ابتدا باید با مفهوم زیر برای توابع ناشناس در متلب آشنا شد:

یک تابع ناشناس تابعی است که در پرونده برنامه ذخیره نمی شود ، اما با متغیری مرتبط است که نوع داده آن function_handle است. توابع ناشناس می توانند چندین ورودی را بپذیرند و یک خروجی را برگردانند. آنها فقط می توانند یک عبارت اجرایی تنها داشته باشند.

اکنون که با توابع ناشناس در متلب آشنا شدیم باید بدانیم که برای ایجاد یک function_handle ، باید نام تابع را با علامت @ قرار داد.

خروجی قطعه کد گفته شده، بصورت مقابل خواهد بود:

```
0.4500
```

همانگونه که انتظار داشتیم، خروجی با مقدار واقعی برابر نیست و این اختلاف ناشی از خطای محاسبه بدلیل استفاده از مقدار متفاوتی برای ورودی تابع میباشد. با تغییر مقادیر ورودی در بازه ای کوچک حول همان مقدار ورودی داده شده در متن سوال، میتوان به نتایج جالبی رسید:

قطعه کد جدید برای مشاهده مقادیر مختلف تابع به ازای ورودی های متفاوت بصورت زیر نوشته شده است:

```
f = @(x) (1-cos(x))/x^2;
for i = -9:0.5:-0.5
    a = pi*10 ^ (i);
    fprintf('f(pi*10^(%.1f)=%f) = %.4f\n', i, a, f(a));
end
```

خروجی این قطعه کد بصورت زیر خواهد بود:

```
f(pi*10^(-9.0)=0.000000) = 0.0000
f(pi*10^(-8.5)=0.000000) = 0.0000
f(pi*10^(-8.0)=0.000000) = 0.4500
f(pi*10^(-7.5)=0.000000) = 0.4950
f(pi*10^(-7.0)=0.000000) = 0.4995
f(pi*10^(-6.5)=0.000001) = 0.5000
f(pi*10^(-6.0)=0.000003) = 0.5000
f(pi*10^(-5.5)=0.000010) = 0.5000
f(pi*10^(-5.0)=0.000031) = 0.5000
f(pi*10^(-4.5)=0.000099) = 0.5000
f(pi*10^(-4.0)=0.000314) = 0.5000
f(pi*10^(-3.5)=0.000993) = 0.5000
f(pi*10^(-3.0)=0.003142) = 0.5000
f(pi*10^(-2.5)=0.009935) = 0.5000
f(pi*10^(-2.0)=0.031416) = 0.5000
f(pi*10^(-1.5)=0.099346) = 0.4996
f(pi*10^(-1.0)=0.314159) = 0.4959
f(pi*10^(-0.5)=0.993459) = 0.4602
```

ابتدا به مقادیر متفاوت از ورودی تابع دقت کنید. همانطور که میبینیم، هنگامی که ورودی بسیار کوچک باشد، خروجی با خطا همراه است و مقدار خطا به دلیل قابل مقایسه بودن با مقدار ورودی قابل چشم پوشی نیست، اما با افزایش مقدار ورودی تا عدد مشخصی، این خطا آنقدر ناچیز میشود که میتوان مقدار تابع را برابر با مقدار واقعی دانست. در واقع چون مقدار خطا دیگر قابل مقایسه با مقدار ورودی نیست میتوان از آن صرف نظر کرد.

بنابراین همانطور که اشاره شد برای اینکه مقدار عددی خطای کمتری داشته باشد میتوان از ورودی های مشخص شده در خروجی قطعه کد بالا استفاده کرد.

بخش دوم:

قبل از محاسبه و نوشتن کد در متلب، از لحاظ محاسبه عددی، کاملا مشهود است که دو تابع زیر با هم یکسان نیستند.

$$\frac{1000^x}{e^{20x}} \neq \left(\frac{1000}{e^{20}}\right)^x$$

برای اثبات این ادعا با استفاده از دستورهای متلب، بصورت زیر عمل میکنیم:

```
f1 = @(x) (1000^x)/(exp(20*x));
f2 = @(x) (1000/exp(20))^x;
a = 200;
fprintf('f1(%d) = %.2f\n', a, f1(a));
fprintf('f2(%d) = %.2f\n', a, f2(a));
```

خروجی قطعه کد بالا بصورت زیر است:

```
f1(200) = NaN
f2(200) = 0.00
```

همانطور که انتظار میرفت، مقادیر با هم متفاوت هستند. اما این اختلاف از کجا ناشی میشود؟

فرض کنید میخواهیم عبارت سمت چپ را محاسبه کنیم، برای اینکار ابتدا هر کدوم از عبارت های صورت و مخرج را حساب میکنیم و سپس حاصل تقسیم آنها را بدست می آوریم. اما مشکل اینکار کجاست؟ در واقع هر یک از عبارات صورت و مخرج، عبارات بسیار بسیار بزرگی هستند و قادر به تقسیم کردن آنها نیستیم. اما وقتی میخواهیم حاصل عبارت سمت راست را محاسبه کنیم، ابتدا حاصل تقسیم را (که عددی مثبت و کوچکتر از یک است) بدست می آوردیم و سپس آنرا به توان میرسانیم که خروجی نهایی عددی بسیار بسیار کوچک و برابر با صفر خواهد بود.

بخش سوم: پردازش تصویر

```
clc; clear;
I = imread('Image.jpg');
subplot(1,2,1);
imshow(I);
title('Original Image')
I2 = I;
[a, b] = size(I);
for n = 1:1:a
    for m = 1:1:b
        if I(n, m) > 130
            I2(n, m) = 1.2*I(n, m);
        else
            I2(n, m) = 0.8*I(n, m);
        end
    end
end
subplot(1,2,2);
imshow(I2);
title('Modified Image')
```

خروجی قطعه کد بصورت زیر خواهد بود: