

report 1

马昕宇 2017011455

实验一

实验要求

用MATLAB编程实现例1.4，绘出图1.2，体会两种误差对结果的不同影响

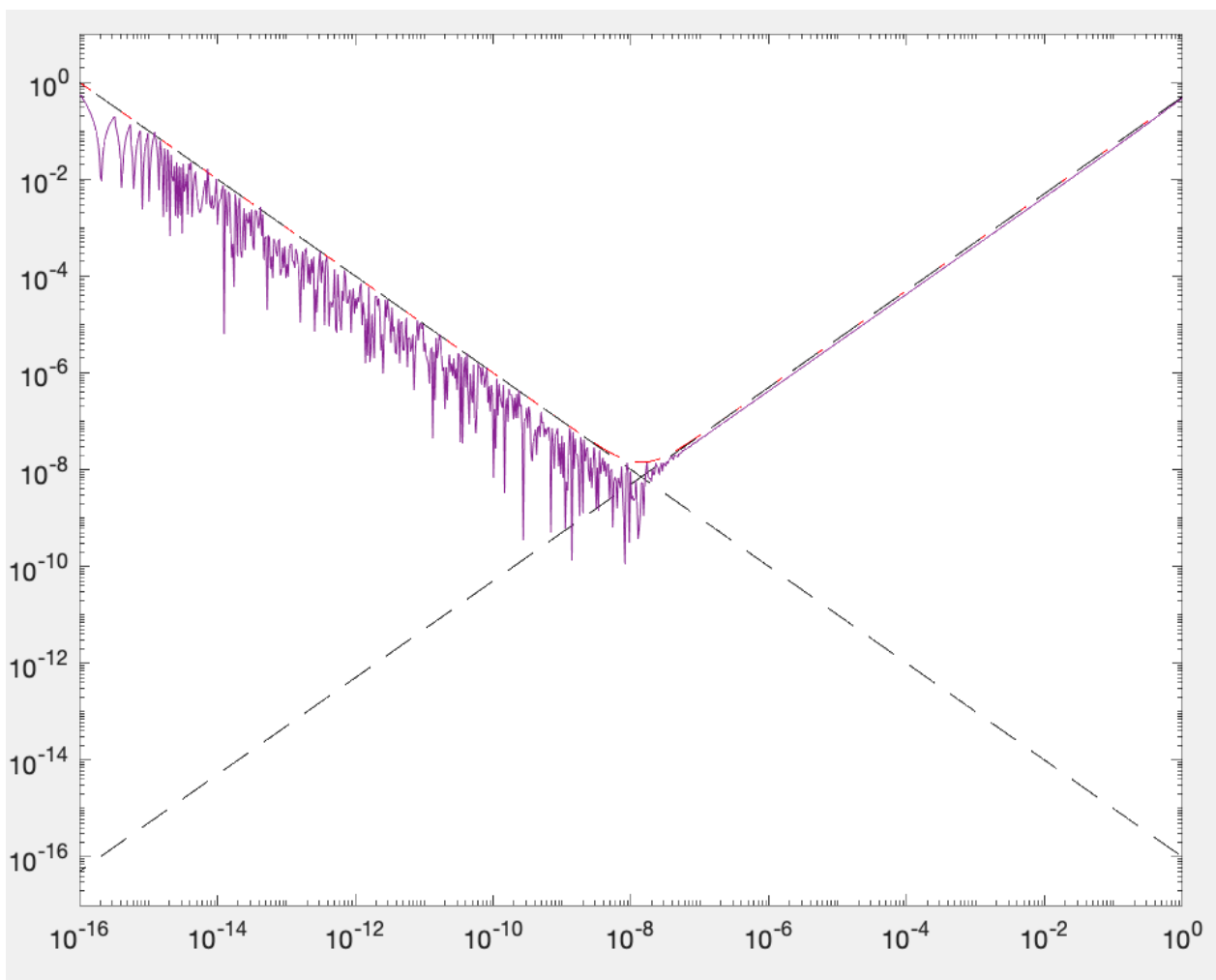
算法实现

只需要按照书中公式将各种误差计算出来即可，代码如下：

```
h = logspace(-16,0,1000);
truncation = h/2;
rounding = 0.0000000000000001 * h.^-1;
epsilon = truncation + rounding;
error = abs(((sin(1 + h) - sin(1)) ./ h) - cos(1));
plot(h, epsilon, h, truncation, h, rounding, h, error), loglog(h, epsilon, '--r', h,
truncation, '--k', h, rounding, '--k', h, error), axis([0.0000000000000001 1
0.0000000000000001 10]);
```

实验结果

绘制出图像如下：



实验二

实验要求

编程观察无穷级数

$$\sum_{n=1}^{\infty} \frac{1}{n}$$

的求和运算.

1. 采用IEEE单精度浮点数,观察当n为何值时求和结果不再变化,将它与理论分析的结论进行比较(注:在MATLAB中可用single命令将变量转成单精度浮点数)。
2. 用IEEE双精度浮点数计算(1)中前n项的和,评估IEEE单精度浮点数计算结果的误差。
3. 如果采用IEEE双精度浮点数,估计当n为何值时求和结果不再变化,这在当前做实验的计算机上大概需要多长的计算时间?

算法实现

1. 记录求和值与上一次循环时的值,循环直到两个值的差为零为止,代码如下:

```

sum = single(1);
last = single(0);
j = 1;

while sum - last ~= 0
    last = sum;
    j = j + 1;
    sum = sum + 1/j;
end

disp(j);

```

2. 与(1)类似，只要去掉single即可:

```

ssum = single(1);
slast = single(0);
dsum = 1;
j = 1;
i = 1;

while ssum - slast ~= 0
    slast = ssum;
    j = j + 1;
    ssum = ssum + 1/j;
end

while i ~= j
    i = i + 1;
    dsum = dsum + 1/i;
end

disp(ssum)
disp(dsum)
disp((ssum - dsum) / dsum);

```

实验结果

1. 运行代码 `lab1_3_1.m`，结果如下:

```

>> lab1_3_1
    2097152

```

即当n=2097152时结束

2. 运行代码 `lab1_3_2.m`，结果如下:

```

>> lab1_3_2
    15.4037

    15.1333

    0.0179

```

相对误差为0.0179

3. 双精度浮点数最多有16位有效数字，因此当 $\frac{1}{n} = 5 * 10^{-16}$ 时，求和结果不在变化，此时 $n = 2 * 10^{15}$ ，实验用的计算机频率为2.8GHz，因此大约需要 $2 * 10^6$ 秒,约为23天。