

# SOLUTIONS

## Introduction to University Mathematics 2018 MATLAB WORKSHEET IV

Sample code.

```
1 %This script evaluates pi using the Gregory series
2 N = input('Enter number of terms : ');
3 A = ones(1,N);
4 A(2:2:N) = -1; % This creates [1 -1 1 -1 ...]
5 B = [1:2:2*N-1]; % This creates [1 3 5 7...]
6 C=A./B; % Ratio of the above two lines
7 x = 4.0*sum(C);
8 fprintf('Estimate of pi using %d term in Gregory series = %.10f \n',N,x)
9 err = 100*(x-pi)/pi;
10 fprintf('Percentage error = %f %% \n', err)
```

Note that in line 4, we are simply replacing all the even entries by  $-1$ .

Alternatively, replace lines 3,4,5 6 above with the following:

```
A=[1:N];
C=(-1).^(A+1)./(2*A-1);
```

Note that  $C$  is simply the expression for each term in the series as given in the question.

I tested the performance of these two versions. With  $\sim 100$  terms they take about the same amount of time. However, with  $\sim 10000$  terms, the first method was roughly a few times faster.

This is because we're performing many unnecessary calculations in the second method to get the alternating signs (namely  $(-1)^n$ ) - this could have easily been put in by hand.

Yet another idea is to evaluate the odd and even sums separately.

```
C= sum(1./[1:4:2*N - 1]) - sum(1./[3:4:2*N - 1]);
```

Always break a coding problem down into several mini-stages, and annotate your solution to each stage.

Here's what NOT to do.

```
x=4*sum((-1).^([1:N]-1)./(2*[1:N]-1))
```

Whilst the code is correct, it is difficult to decipher what the heck is going on in the author's head.

Other common mistakes include:

- forgetting to suppress unnecessary display of arrays that mess up the user's screen.
- using too many unnecessary brackets and getting confused by them
- using `%d` instead of `%f`. The percentage error should be a float, not an integer.

**Marking scheme** (10 marks in total)

**10 marks** = A perfectly working code which runs cleanly. The script is clearly laid out and the thought processes are clearly communicated via annotation.

The following errors will cost you one mark each.

- No useful annotation
- Series evaluation doesn't contain 10 decimal places
- Wrong kind of conversion character used (e.g. if `%d` is used to report the percentage error)
- My screen is messed up after running your code (e.g. `\n` or `;` forgotten)
- The percent sign isn't displayed using `%%`
- Other programming errors the marker considers unacceptable

However, If your code gives the correct numerical answers with no serious error, the minimum you will get is 5/10.

**Serious errors**

- If the code runs but any of the answers is wrong, the maximum is 5/10.
- If the code does not run, the maximum is 4/10.