Ojas Bardiya
UID: 505145284
CEE/MAE 20
07/03/2020

1. Calculating the Surface Area of an Oblate Spheroid

1.1. Introduction

We calculate the surface area of an oblate spheroid using the given formula and comparing the result we obtain to the method used to approximate the same.
We used the built-in MATLAB π and the built-in trigonometric functions cos and sin for cosine and sine respectively. We also use the built-in MATLAB functions acos and log for arc cosine and natural logarithm (ln) respectively.

1.2. Model and Theory

We use the equation –

$$A(r_1, r_2) = 2 * \pi * (r_1^2 + \frac{r_2^2}{\sin\gamma} * (\log\frac{\cos\gamma}{1 - \sin\gamma}))$$

where $\gamma = \cos^{-1}(\frac{r_2}{r_1})$
and the approximation –

$$A = 4 * \pi * ((r_1 + r_2)/2)^\wedge 2$$

And find the discrepancy between the 2.

1.3. Methods and Pseudocode

<u>Pseudocode:</u>

```
Allow the user to input both values of r1 and r2
Check if the values entered are valid by making sure that r2 < r1 and both r1 and r2 are positive real numbers
Calculate the Surface area using both the formula and approximation.
Output both values using the fprintf function
Calculate the discrepancy upto 10 digits and output it
```

1.4. Calculations and Results

For the input (6378.137, 6356.752) we have –

```
Output:
The surface area is: 5.100656e+08
The approximation is: 5.094953e+08
The discrepancy is: 5.702833e+05
```

For the input (34.55, 36.77) we have –

```
Output:
The equatorial radius must be larger than the polar radius!
```

For the input (-10, 45.67) we have –

```
Output:
Both radii must have a non-negative real value!
```

1.5. Conclusion

Clearly, we can see that there is a discrepancy between the two values, indicating that for larger values the approximation becomes less and less accurate.

2. Neighbor identification through linear indexing

2.1 Introduction

We find all neighbors of a cell P in a 2-D M*N grid and output all of them.

2.2 **Model and Theory**

Consider a given cell P. Its neighbors are given by –

```
P_up = P - 1;
P_down = P + 1;
P_right = P + M;
P_left =  P - M;
%upper left diagonal
P_lu = P - M - 1;
%upper right diagonal
P_ru = P + M - 1;
%lower left diagonal
P_ll = P - M + 1;
%lower right diagonal
P_rl = P + M + 1;
```

There are 3 types of cells –

- Interior – These have 8 neighbors, which is the maximum
- Edge – These have 5 neighbors
- Corner – These have 3 neighbors, which is the minimum

We determine the type of cell and output the its neighbors accordingly.

2.3 Methods and Pseudocode

<u>Pseudocode:</u>

```
Allow the user to input the number of rows, columns and cell ID as M, N and P respectively
Make sure that each value entered is valid
Determine the cell type and output its neighbors accordingly using their relation to the cell ID of P as described in the model
```

2.4 Calculations

For (M, N, P) = (4,6,12) we have –

```
Output:
Cell ID:       12
Neighbors: 7  8  11  15  16
```

2.5 **Conclusion**

We can determine the neighbors of a cell simply by knowing its index as there is relationship between each of its neighbors depending on the values of M and N.