

Processing Ionospheric Data Using MATLAB

Daniel Okoh

NASRDA Center for Atmospheric Research

African School on Space Science, July 2014

Licensing

Recently, various legal free copies of the software from Mathworks

Modeling and Simulation using MATLAB on iversity (<https://iversity.org>)



INTERDISCIPLINARY

Modelling and Simulation using MATLAB®

Prof. Dr.-Ing. Georg Fries, Prof. Dr. Peter Dannenmann, Prof. Dr. Karin Graeslun...

How can I simulate a water treatment plant or realize a new business venture? This MOOC explains how to model and simulate innovative ideas using MATLAB/Simulink.

🕒 22 APR. 2014 🗨️ ENGLISH 📌

GO TO COURSE

Outline

Calling the IRI Model from MATLAB
(An offline MATLAB version of the IRI Model)

Reducing GPS Data on MATLAB

Data Visualization on MATLAB
(Spatial & Temporal)

A bit of IRI Intro

The IRI is acronym for International Reference Ionosphere;

➤ International Standard for Specifying Ionospheric Parameters (Bilitza, 2001)

Model is developed empirically

Parameters include electron densities, plasma frequencies, peak heights and densities, TEC, etc.



IRI Availability through IRI Homepage (<http://irimodel.org/>)

Online Computation

Fortran source code

**MATLAB Version (By Drew Compston);
Requires internet; as it works with the online
interface**

A MATLAB version that does not require internet

**Utilizes the Fortran source code, and so does not
require internet**

**Any suitable Fortran compiler can be used to
compile the IRI source code; I use
simplyFortran (<http://simplyfortran.com/>)**

A MATLAB version that does not require internet

The screenshot shows the IRI2012Matab MATLAB interface. The window title is "IRI2012Matab". The interface is divided into several sections:

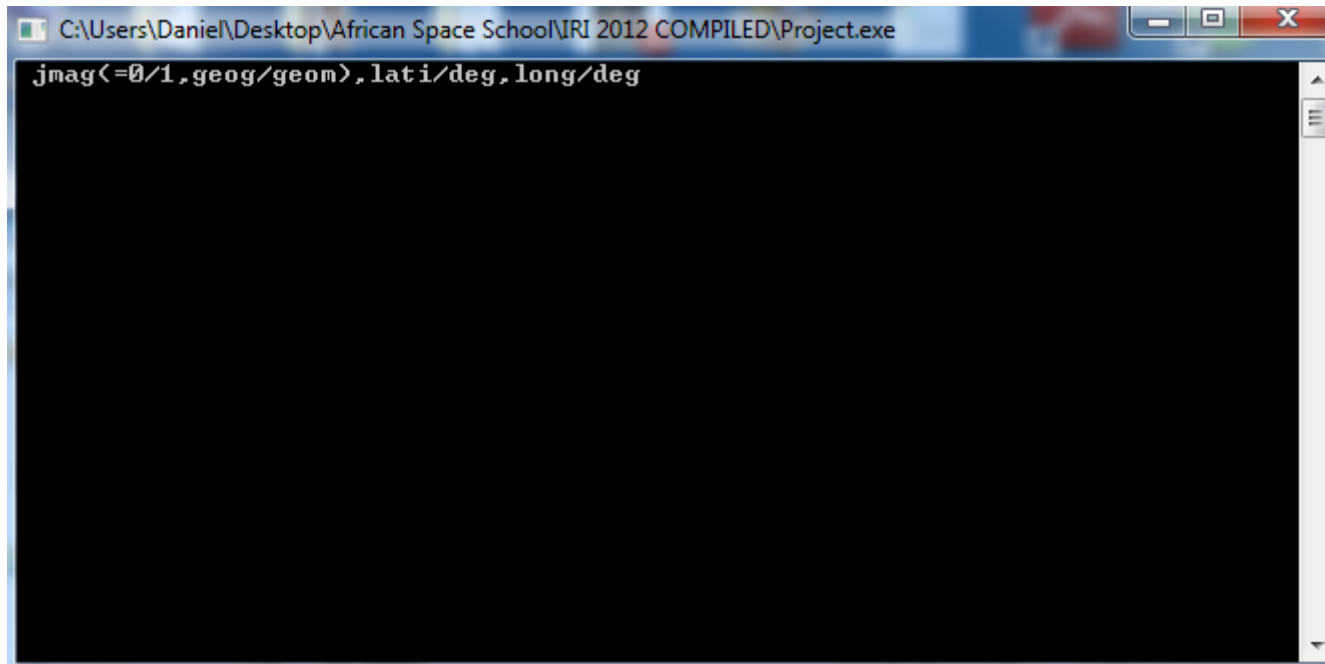
- Time and Location:** Year (1958 - 2016) set to 2000; Month set to January; Day (1 - 31) set to 1; Time set to Universal; Hour of Day (e.g. 1.5) set to 1.5.
- Coordinates:** Coordinates Type set to Geographic; Latitude (deg., from -90 to 90) set to 7.5; Longitude (deg., from 0 to 360) set to 7.2; Height (km, from 60 to 2000) set to 100.
- Profile Parameters:** Select a Profile type and its parameters: Height, km [60 to 2000]; Start set to 100; Stop set to 2000; Stepsize set to 50.
- Optional (for TEC computation):** No Topside checked; NaQuick set to NaQuick; TEC Upper boundary (km, from 50 to 2000) set to 1000.
- Select desired IRI output parameters:** A list of radio buttons with the first option selected: "I want a Standard table of default IRI parameters". Other options include: "I want a List of peak heights and densities", "I want a List of plasma frequencies, D0, M3000, valley, width and depth", "I want a List of 6 parameters of my choice", and "I want a List of D-region models at 60, 65, ..., 110 km".
- Submit:** A large button at the bottom center.

Text on the right side of the output selection section: "This option will give you a standard table of default IRI model parameters which usually include the following: Electron Densities (Ne(3000) and Ne(NmF2)), Temperatures (Tm3000, Tm and Te(K)), Ion percentages (O+, N+, H+, He+, O2+, NO+), TEC and TEC by percentage."

← Outside ILLUSTRATION →

Stages/Considerations

Ordinarily, a compiled IRI Fortran code will require user inputs on the command prompt; could be strenuous to generate a single profile



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\Daniel\Desktop\African Space School\IRI 2012 COMPILED\Project.exe. The command prompt displays the following text: `jmag(=0/1,geog/geom),lati/deg,long/deg`. The rest of the window is black.

← Outside ILLUSTRATION →

Stages/Considerations

Modification of the fortran code before compilation:

- **the program should read inputs from a specified file rather than wait for the user inputs on the command prompt**

MATLAB file processing used to write user inputs from the GUI to the specified file

The compiled IRI program is called using the MATLAB 'dos' command

MATLAB file processing used to read IRI data from the IRI output file

Reading and Writing to Files on MATLAB

Writing:

```
fid=fopen('filename.txt', 'wt');
```

```
a=magic(7)
```

```
fprintf(fid, '%f\t %f\t %f\t %f\t %f\t %f\t %f\n', a');
```

Reading:

```
fid=fopen('filename.txt');
```

```
b=textscan(fid, '%f\t %f\t %f\t %f\t %f\t %f\t %f\n')
```



Outside ILLUSTRATION

Multiple Profiles

Example; Daily profiles for a whole year



GPS TEC from GPS Observations & RINEX Files

A MATLAB RINEX2TEC converter

**Elaborate treatment by John Raquet & Luigi
Ciraolo next week**

**The ionosphere delays GPS signals by amounts
that depend on the signal frequencies.**

**By receiving at 2 different frequencies, we can
estimate these delays, which in turn correspond to
some value of TEC**

Reducing GPS TEC Data (A Case of SCINDA GPS)

WinTEC-P computes TEC from RINEX files; produces compressed (.gz) files

```
gunzip('* .gz'); %to uncompress all GNU zip files  
to a new folder named 'output' in the current  
folder
```

```
find(vtec~=-1 & elev>=20); % to exclude vtec  
data = -1.00 and satellite elevations less than 20  
degrees
```

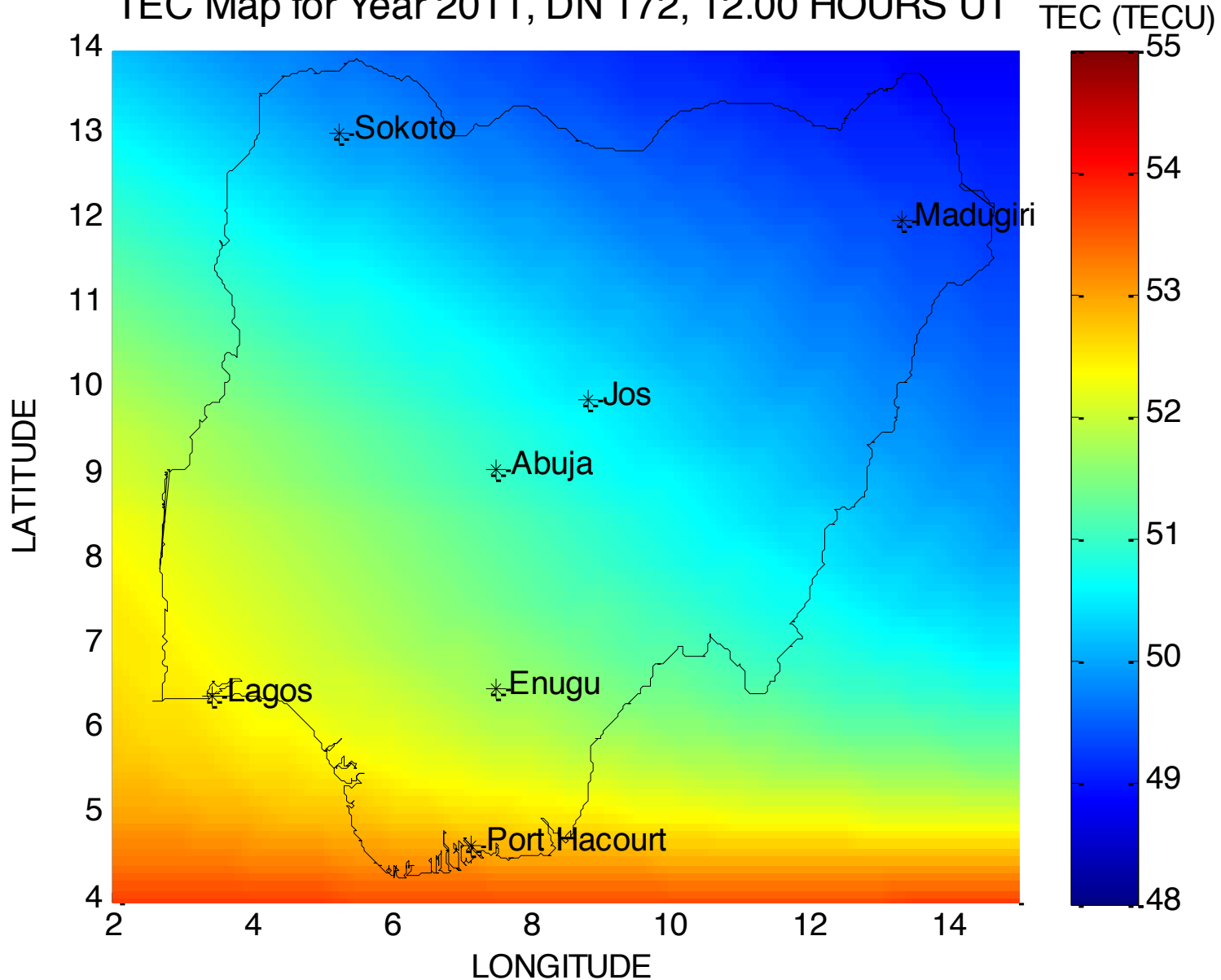
```
find(yy==y & mm==m & dd==d &  
hfrac==round(h)); % y is year, m is month, dd is  
day of month, hfrac is hour fraction
```

Data Visualization

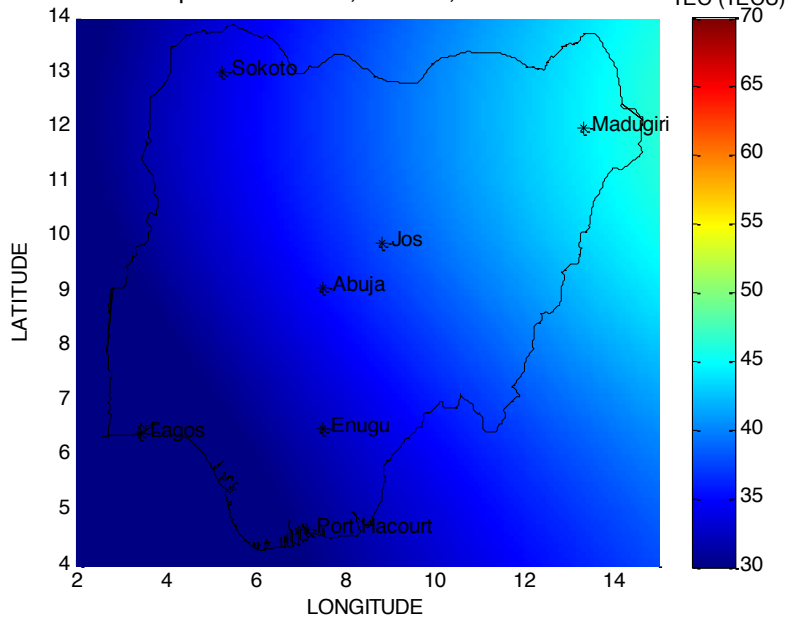
Spatial Variations and Temporal Variations

Spatial Variations

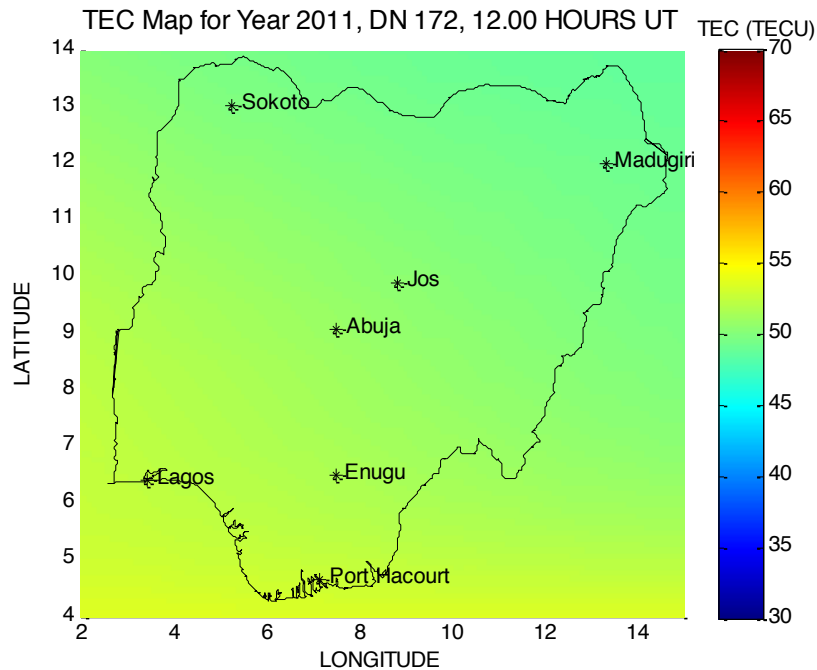
TEC Map for Year 2011, DN 172, 12.00 HOURS UT



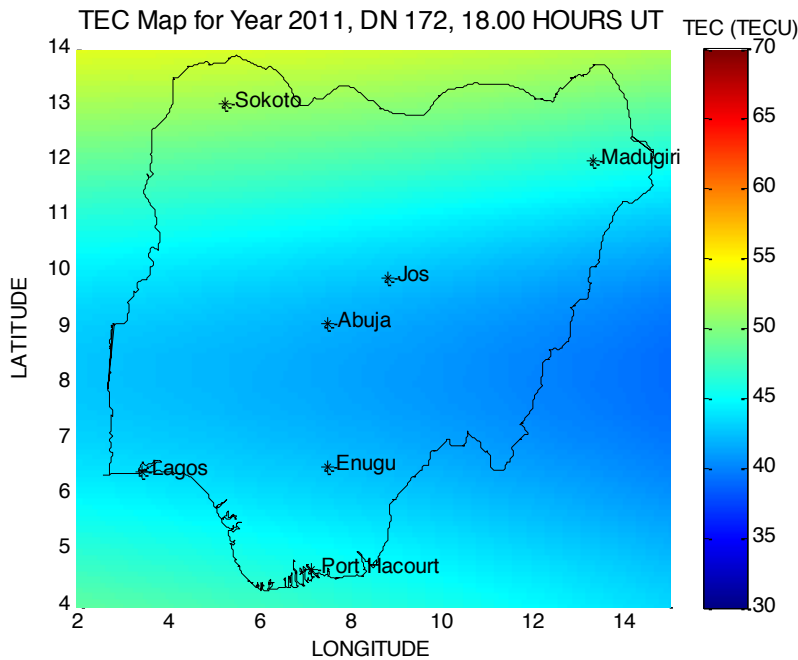
TEC Map for Year 2011, DN 172, 6.00 HOURS UT



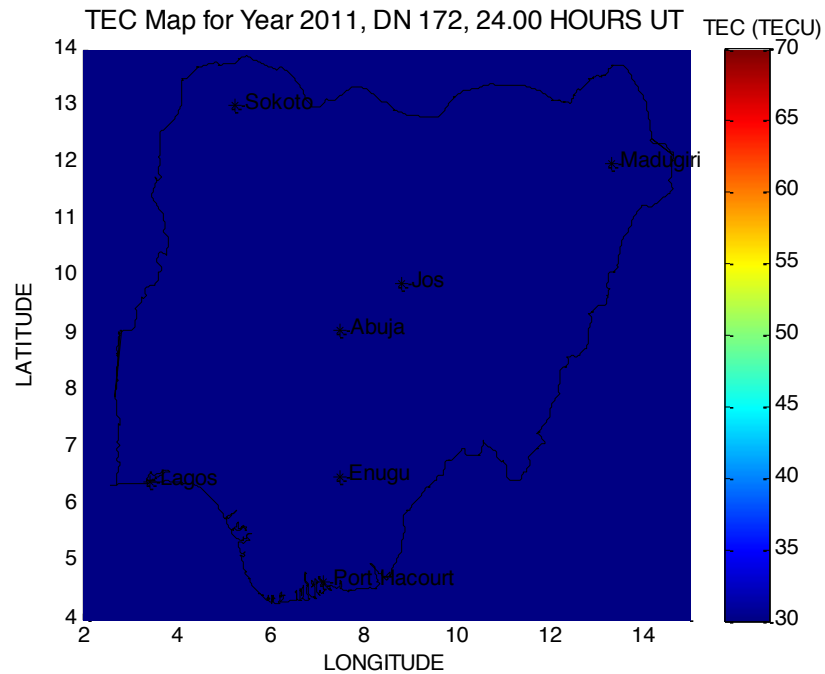
TEC Map for Year 2011, DN 172, 12.00 HOURS UT



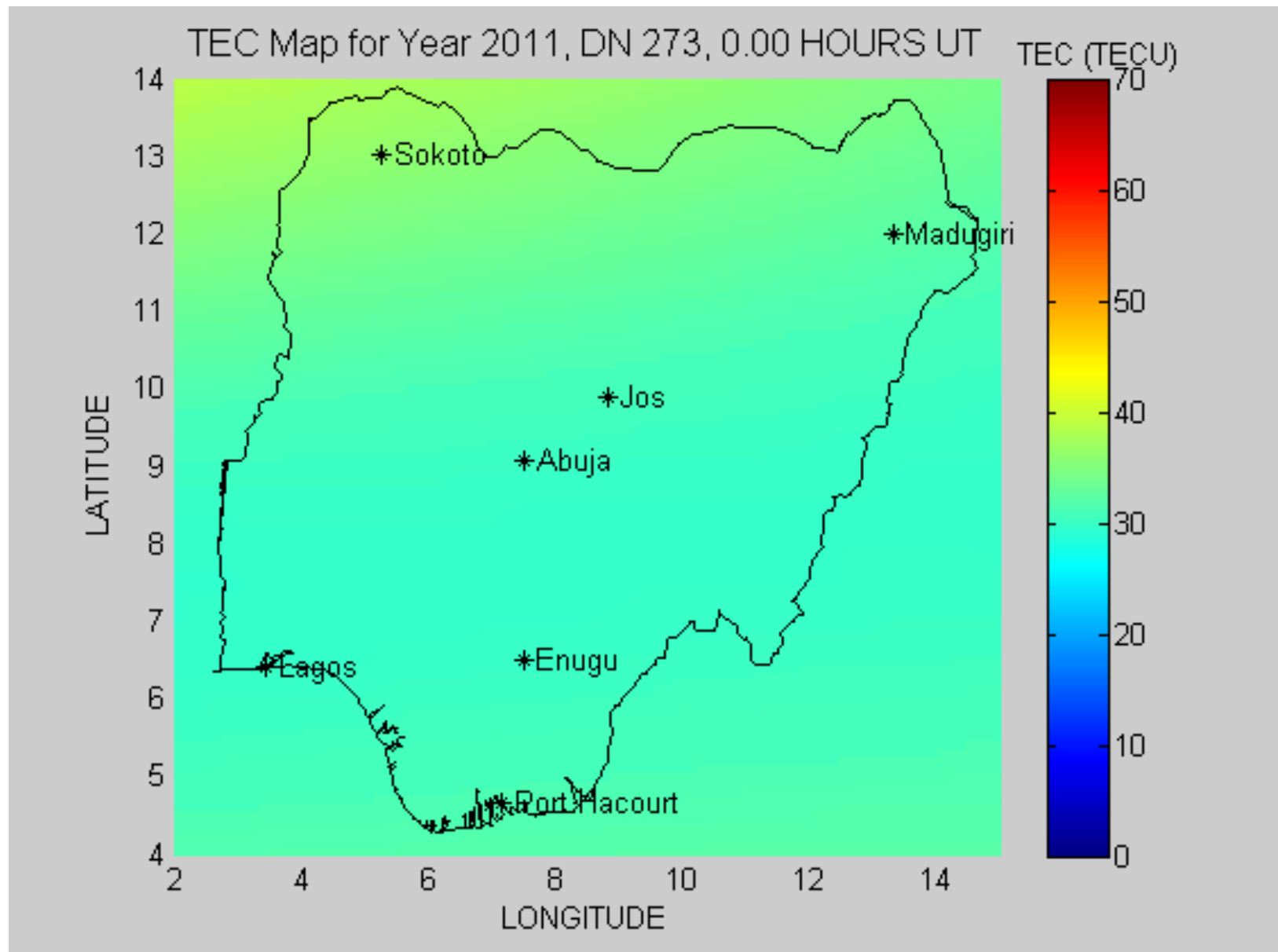
TEC Map for Year 2011, DN 172, 18.00 HOURS UT



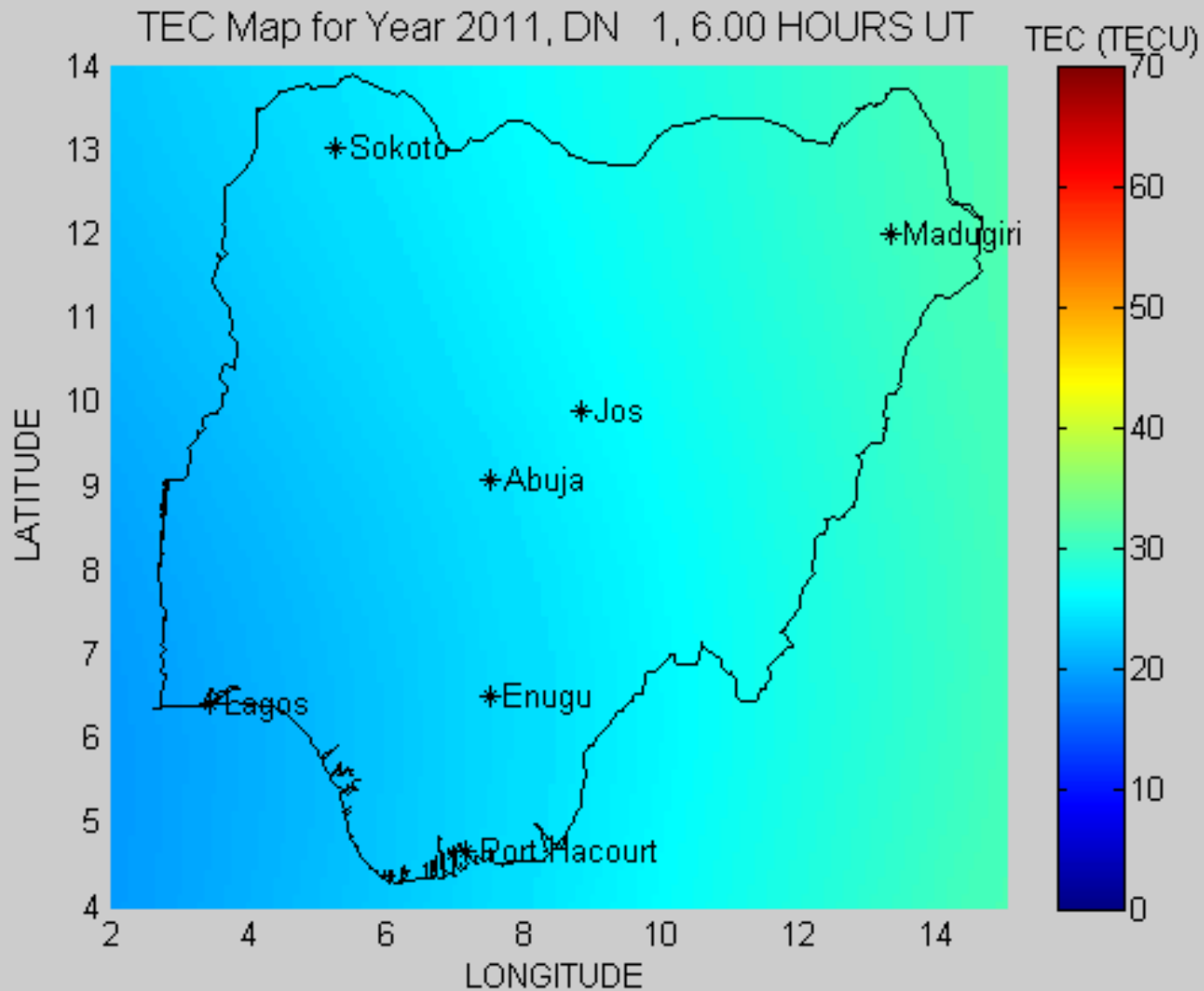
TEC Map for Year 2011, DN 172, 24.00 HOURS UT



Spatial and Temporal



Spatial and Temporal 2



MATLAB Script; Spatial

imagesc

imagesc(tec)

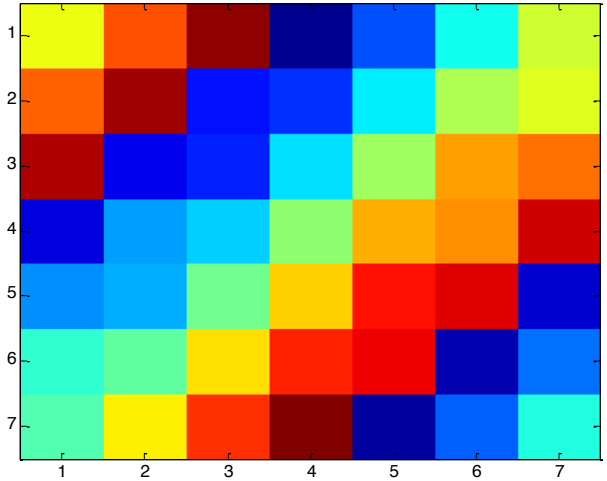
Example tec=

```
[30  39  48  1  10  19  28  
38  47  7  9  18  27  29  
46  6  8  17  26  35  37  
5  14  16  25  34  36  45  
13  15  24  33  42  44  4  
21  23  32  41  43  3  12  
22  31  40  49  2  11  20];
```

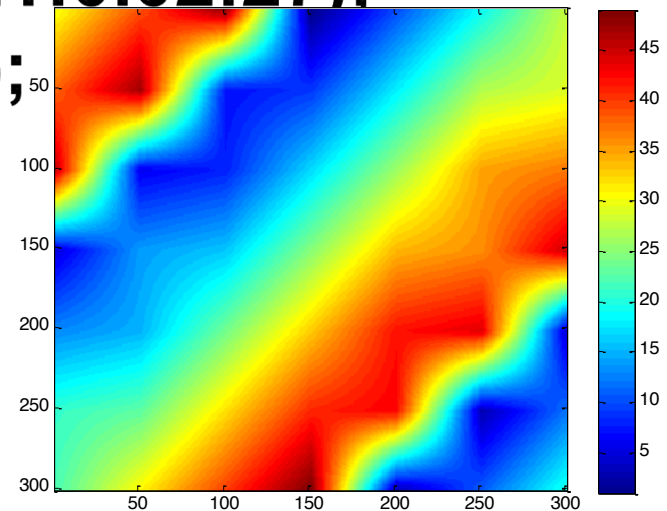
MATLAB Script; Spatial

```
a=magic(7);  
imagesc(a);
```

29	46	6	8	17	26	35
37	5	14	16	25	34	36
45	13	15	24	33	42	44
4	21	23	32	41	43	3
12	22	31	40	49	2	11
20						

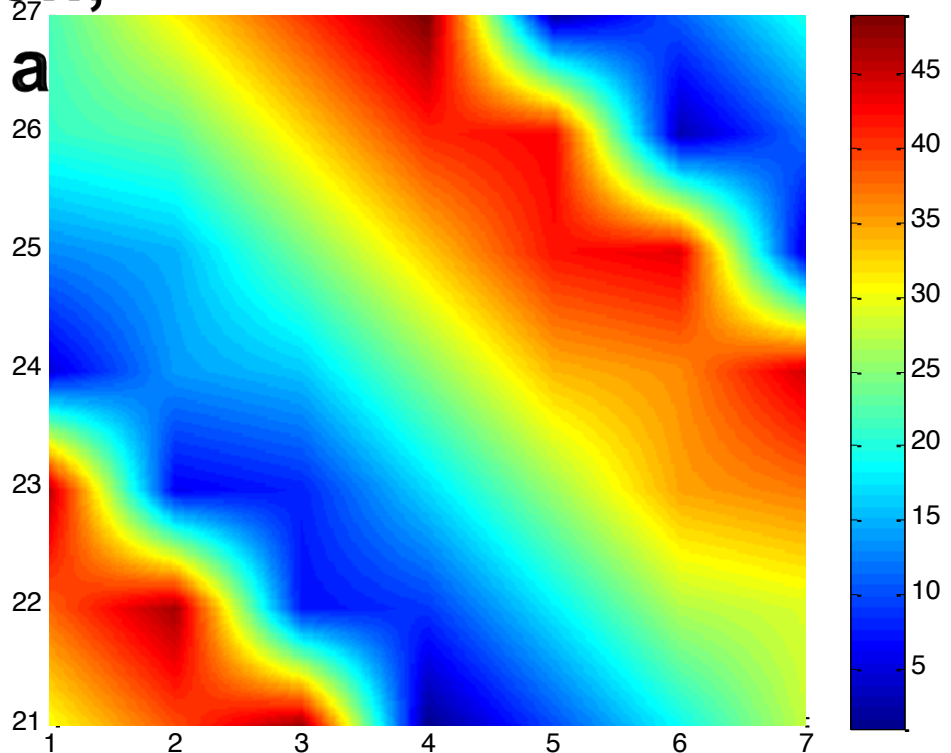


```
a=magic(7);  
[long lat]=meshgrid(1:7, 21:27);  
[long2 lat2]=meshgrid(1:0.02:7, 21:0.02:27);  
a2 = interp2(long,lat,a,long2,lat2);  
imagesc(a2);  
colorbar;
```



MATLAB Script; Spatial

```
a=magic(7);  
[long lat]=meshgrid(1:7, 21:27);  
[long2 lat2]=meshgrid(1:0.02:7, 21:0.02:27);  
a2 = interp2(long,lat,a,long2,lat2);  
axis([1 7 21 27]); hold on;  
imagesc([1 7], [21 27], a)  
colorbar;
```



MATLAB Script; Spatial & Temporal

```
aviobj = avifile('filename.avi');
```

```
for t=1:n
```

Program to make each figure here

```
F = getframe(gcf);
```

```
aviobj = addframe(aviobj,F);
```

```
end
```

```
aviobj = close(aviobj);
```



```
for t=1:50
```

```
    a=magic(7)-t;
```

```
    [long lat]=meshgrid(1:7, 21:27);
```

```
    [long2 lat2]=meshgrid(1:0.02:7,  
21:0.02:27);
```

```
    a2 = interp2(long,lat,a,long2,lat2);
```

```
    axis([1 7 21 27]); hold on;
```

```
    imagesc([1 7], [21 27], a2);
```

```
    caxis([-10 60]); colorbar;
```

```
    title(['t=' num2str(t)]); hold off;
```

```
    F = getframe(gcf);
```

```
    aviobj = addframe(aviobj,F);
```

```
end
```

```
aviobj = close(aviobj);
```

Thank you