Rudimentary Creation and Visualization of Tissot's Indicatrix for Mercator and Cylindrical Equal Area Projection

Esmé Middaugh
March 7, 2022
Cartographic Software Adaptation Final Project

Task Description

Given maximum latitude and longitude, a desired grid spacing, and a projection, visualize Tissot's Indicatrix. After consulting with Dr. Prechtel, this task was broken into three parts. First, create the desired graticules in latitude and longitude. Second, create projection functions that transform the latitude and longitude coordinates to x and y coordinates. Third, find the scale factors for appropriate sizing of the indicatrix.

I sincerely tried to implement this with derivatives to make it reusable for multiple projections. After hours (truly) reviewing basics of partial derivatives and calculus, nomencalture I gave up. My trigonometry and calculus isimply too poor to fully understand explanation given in the "Map Projections: A Working Guide." Given this, I focused only on Mercator and Equal Area, as I could find the already derived equations for the scale factors and they did not require rotation.

Additional Note

I have also included the original script that I wrote that relied more heavily on packages but that has an output more similar to what we originally discussed. A few screenshots of its output are included at the bottom of this report..

Relevant external sources and approaches

For this project, my key source was a post on stack overflow on 'How to Create and Accurate Tissot's Indicatrix' and the source mentioned there, 'Map Projections: A Working Manual.' I also consulted Dr. Prechtel multiple times, and read "Measures and visualization methods of map projection distortions with the use of "python matplotlib library" as an example." Another approach that approximates Tissot's Indicatrix is described in Esri's Making the Tissot's Indicatrix. This approach uses buffers of set kilometers to create the polygons, as their shape and size get distorted in an approximation of an Indicatrix. This is a similar approach to that taken by Mike Bostock in his D3 Block on Tisstot's Indicatrix.

Algorithm(s) used & Modularisation of the implemented version

Base python packages used included numpy and matplotlib. Key functions / algorithms include :

create_graticules - Creates a grid in the latitude longitude space of intersecting meridians and parallels, spaced according to the given merdian and parallel spaces.
plot_grid - plot the graticules
obtain_indicatrix_axis - For a given grid and projection, return the vertical and horizontal axis scaling for and indicatrix at that point
plot_tissots_indicatrix - For a given grid of meridians and parallels and a projection (currenlty only one of 'mercator' or 'cylindrical_equal_area'), create an indicatrix plot with indicatrices scaled to approximately 6% of the graph size.

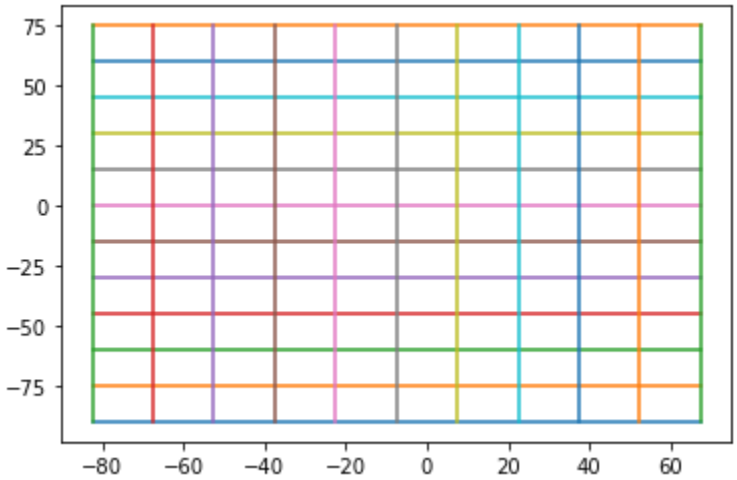
Helper functions
transform_grid - go through a nested list meridians and parallels and return a new list with the points transformed
secant - a shortcut to 1 / np.cos()

Projection Functions
mercator - Given lat and longitude, return the mercerator x and y coordinates
cylindrical_equal_area - Given lat and longitude, return the the cylindrical equal area coordinates

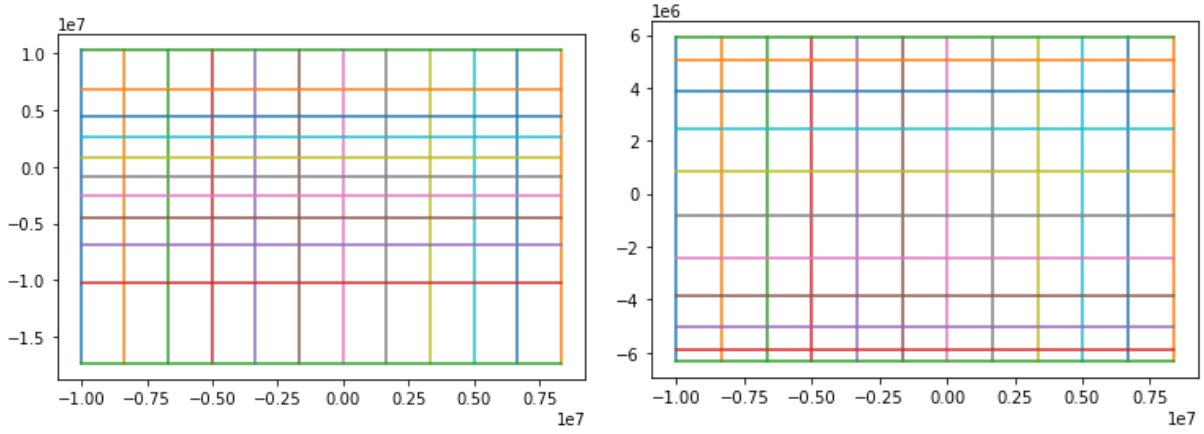
Results

After consulting with Dr. Prechtel I recreated the first two graphs he made, consisting of the initial latitude and longitude graticules and the projected versions.

My first function creates a grid of meridians and parallels



Mercator - Rectangular Conformal and *Cylindrical Equal Area:*

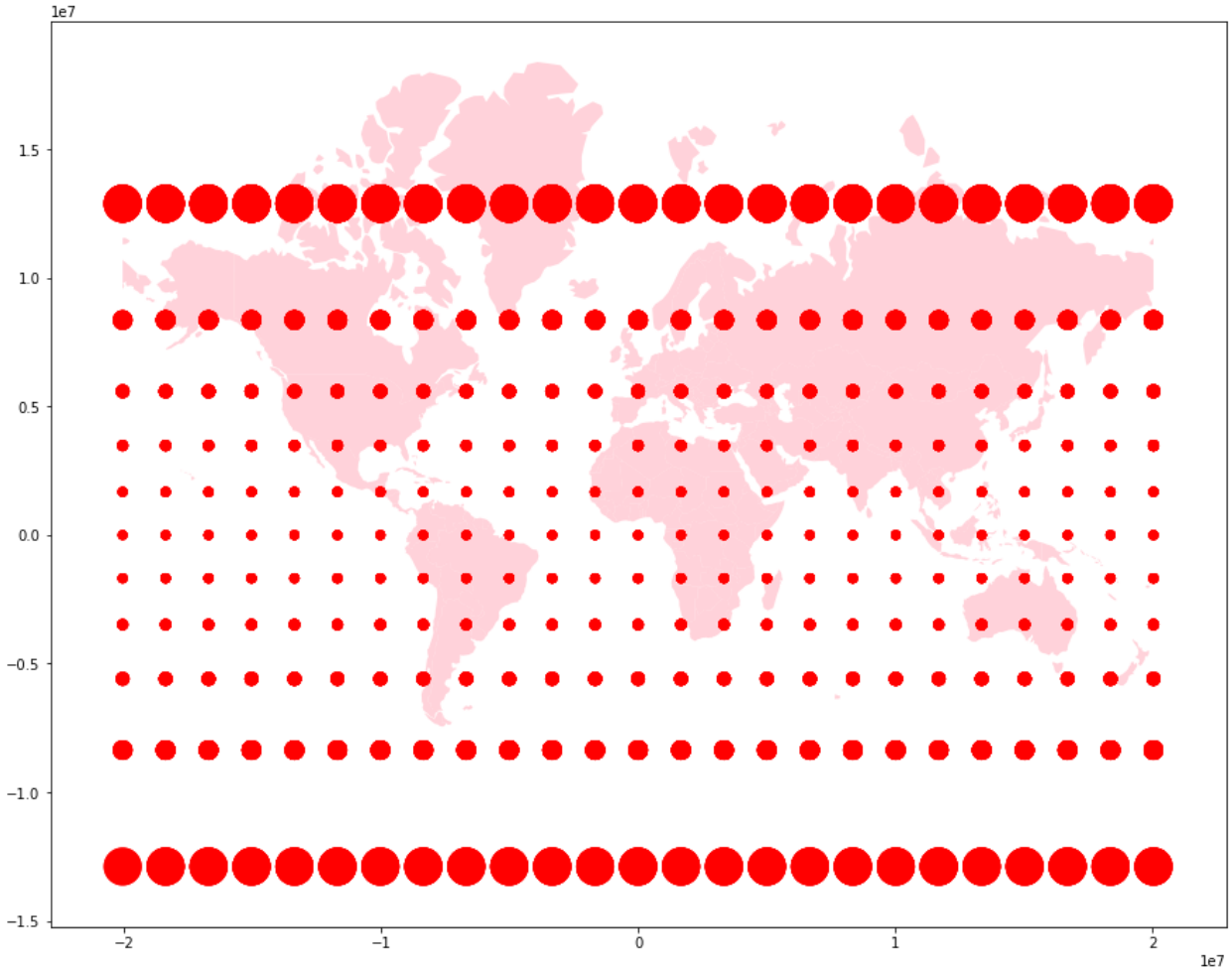


As I could not

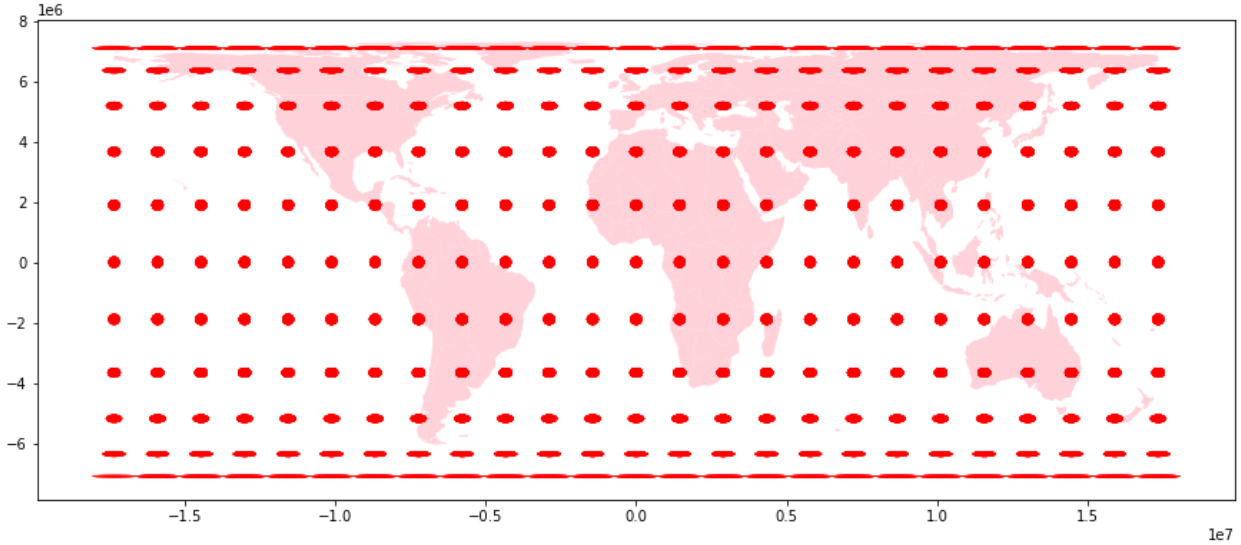
Tissot's Indicatrix Obtained for Mercator and Cylindrical Equal Area:

Below are some examples that I created from my initial package-heavy approach. While I know that this isn't preferable, the final output was a lot closer to what I wanted to have as a final output so I wanted to include it here. This version used a base map from Natural Earth. I created a grid of LineStrings, found the intersection points, and then created the Polygons by making Ellipses. to create and map the indicatrices The script also writes these polygons to a to shapefile for future use.

EPSG 3395 Mercator - Rectangular Conformal



EPSG 6933 (Equal Area)



Future Work

The obvious future work here is to incorporate the general derivative calculation of Tissot's Indicatrix to allow for more projections. I struggled with this for many hours–truly– but my calculus and trigonometry are so poor that even after reviewing numerous videos on partial derivatives, nomenclature, etc. I couldn't get the equations from Snyder's *Map Projections* book to work and ended up settling on just these two that I was able to find equations for the scale factors using just lambda and phi.

Additional future work would be modifying for a ellipsoid instead of a sphere, including projections that would also need an axis of orientation, incorporating a base map and much more.

References

My core references are listed in the Relevant External Sources section. However, in my futile attempt to understand the necessary calculations, I also referenced the presentation "The Mathematics of Maps" (Part One and Part Two). I also reviewed this webpage from kartoweb and referenced the Mercator Wikipedia Page I used for the final simplified version of the scale factors.