Fourier-Series Exercise.

Part I. Fourier sine series.

1. Analytically derive a Fourier sine series that fits the diffusion equation for the following initial condition function:
$$f(x) = \frac{T^*x}{l}$$
which is an analytic expression of a piece of a sawtooth wave. (Physically, this could represent a situation in which the rod of length l had been held at steady state with different temperatures at each end, following which the hot end was brought to the same temperature as the cold end.)

2. Modify program `boxrod.f95` to show (at least) the isochrones of this solution. You may retain the assumptions that $\kappa = T^* = l = 1$. This may be as simple as modifying the Fourier Series definition statement—a `FORALL` near the beginning.

Required Output for Part I

The integration leading to the formula for the Fourier coefficients, and the graph of isochrones of the solution.

Part II. Fourier-Legendre series.

Use both Fourier-Legendre and polynomial regression methods to calculate polynomial expressions for a zonally-averaged climate field.

1. The file `zonal.data` contains a variety of useful climatic parameters for each of the 18 10° latitude bands. The first three columns are the ocean fractions f_O, ocean cloudcover fractions f_{cO}, and land cloud cover fractions f_{cL}. These three of theses have been gridded for you int `zonal.gdata`. The colums of `zonal.gdatat` are $mu = \sin\phi$ (sine of latitude), which are given in 201 evenly spaced values $\Delta\mu = 0.01$), latitude in degrees, f_O fraction of the latitude band occupied by ocean, f_{cO} cloud cover fraction over ocean in the latitude band, and f_{cL} cloud cover fraction of land in the latitude band. Read those in and calculate f_c weighted average cloud cover for the latitude band.
$$f_c = f_O f_{cO} + (1 - f_O) f_{cL}$$
(Calculating f_c can be done in the same loop as reading f_O, f_{cO}, and f_{cL}, after which you are finished with all the input variables except mu and do not need to retain their values in arrays. You will need arrays for mu and f_c in what follows. It will be useful for graphical purposes to have f_c be stored in a two-dimensional array of shape `(201,3)` where the raw data are in `(1:201,1)` and the two calculated versions below will be stored in `(1:201,2:3)`.

2. Calculate a Fourier-Legendre fit to the cloud cover data. You may truncate your series at order 6. (Excellent programming would make the order a parameter that can be adjusted in one place.)

a) Calculate Fourier-Legendre coefficients by numerically integrating the product of the cloud cover data and the Legendre polynomial values over their range. (See note below about `sfsphere.f95` for Legendre functions.) It will be useful for Part 3 if you calculate the polynomial values in advance for all μ values and all orders and save them in a two-dimensional array. One of the purposes of having a large number of μ values is so that any old integration procedure can be used (i.e., trapezoidal is good enough.)

b) Calculate the Fourier-Legendre fit to cloud cover data by summing the series. (Recommended: put this in `(1:201,2)` of your cloud-cover array.)

3. Calculate coefficients for the Legendre polynomials P_0 through P_6 using a multiple regression. The independent variables you feed to a multiple regression routine are P_1 through P_6, because the coefficient of P_0 will be the "constant" or "intercept" term of the regression. Use any multiple regression procedure you like: IMSL's routine is `RLSE` and LAPACK has `GELS`. Calculate the regression-Legendre fit to the cloud cover data by summing the series. (Recommended: put this in `(1:201,3)` of your cloud-cover array.)

Required Output for Part II

A. Generate a graph of "cloud cover fraction" versus "sine of latitude" which contains three curves: the original data and the two Legendre polynomial fits.

B. Print a table comparing the Legendre polynomial coefficients calculated in steps 3 and 4.

Useful Output – not required for this exercise

Tangible output from this exercise is software that can easily interpolate latitudinal values using Legendre functions. You will also need *all* of the fields in `zonal.gdata` interpolated to a fine-mesh grid of μ_i values for the final project, so either document your code well or do the interpolation now. (Order 10 will be preferable for temperature, the rest should be fine with the order 6 version.) Intangible result is the realization that the Fourier fit works the same as the least-squares fit, which is intended to provide a little faith in the basic math behind spectral GCMs.

Resources

The link `~hanson/cld` contains `boxrod.f95` and `zonal.gdata` mentioned in the instructions above. It also contains `sfsphere.f95`, which holds the module `sfsphere` ("special functions on a sphere"), containing three functions. The first of these is useful for this project:

> `Legendre ( n, x )` calculates the Legendre Polynomial value, $P_n(x)$. n is the order, which must be a nonnegative integer. x is the function argument (μ in lecture), which must be in the range $[-1, 1]$.