Downloading and Analyzing Weather Forecast Data with rNOMADS using the GrADS - DODS system

Daniel Bowman

March 22, 2017

Contents

1 Motivation

I designed the rNOMADS package to provide a quick and easy interface to the NOAA Operational Model Archive and Distribution System. It can retrieve data in ascii format (this works for all operating systems) or in a binary format called grib. This document describes downloading, reading, and displaying ascii data using the GrADS-DODS interface. Please refer to the companion document for information on downloading and processing data in GRIB format.

The rNOMADS package provides online access to 55 weather, ocean, and sea ice models via the GrADS-DODS. This vignette shows how to download data from the Global Forecast System 0.5 x 0.5 degree model. This model describes the state of the atmosphere across the entire planet in 0.5 by 0.5 degree cells.

2 Making Global Weather Maps

Here, we make five maps. The first four are for the analysis forecast - the state of the weather at the time the model was run. Figure 1 is the temperature at 2 meters above the ground. High altitude regions such as Tibet and the Andes show up quite well. Figure 2 is the temperature at the 300 mb pressure level - approximately the height of the jet stream. Figure 3 is the relative humidity at 2 meters above the ground. Note how desert regions of the world are clearly depicted. Figure 4 is the wind speed at 300 mb. This is generally where the jet stream shows up. The last figure (Figure 5) is the surface wind gust forecast 6 hours from now. Hurricanes and tropical storms appear as small circular regions in this map.

2.1 Code

```
> #Get model data:
> #Planetary temperature, relative humidity, winds
> #at 2 m above ground and at 300 mb (jet stream level)
>
> library(GEOmap)
> library(rNOMADS)
> library(fields)
> library(aqfig)
> #Get latest GFS model
>
> model.urls <- GetDODSDates("gfs_0p50")
> latest.model <- tail(model.urls$url, 1)
> model.runs <- GetDODSModelRuns(latest.model)
> latest.model.run <- tail(model.runs$model.run, 1)
> #Get data
>
> time <- c(0,0) #Analysis model
> lon <- c(0, 719) #All 720 longitude points
```

```
> lat <- c(0, 360) #All 361 latitude points
> variables <- c("tmp2m", "rh2m")
> ground.data <- DODSGrab(latest.model, latest.model.run,
+    variables, time, lon, lat, display.url = FALSE)
> ground <- ModelGrid(ground.data, c(0.5, 0.5))
> lev <- c(28,28) #get 300 mb level
> variables <- c("tmpprs", "ugrdprs", "vgrdprs")
> atmos.data <- DODSGrab(latest.model, latest.model.run,
+    variables, time, lon, lat, levels = lev, display.url = FALSE)
> atmos <- ModelGrid(atmos.data, c(0.5, 0.5))
>

> #FIGURE 1
> #Temperature at ground level
>
> #Make model grid
>
> #Set up color scale
> colormap <- rev(rainbow(100, start = 0 , end = 5/6))
> ind <- which(ground$variables == "tmp2m")
> #Save image to PNG file
> #Omit this line if you want to display image
> #rather than save to file
> png(file = "fig_ground_temp.png", width = 1000, height = 750)
> #Make forecast image
> image(ground$x, sort(ground$y), ground$z[1,ind,,], col = colormap,
+     xlab = "Longitude", ylab = "Latitude",
+     main = paste("World Temperature at Ground Level:",
+     ground$fcst.date))
> #Plot coastlines
> plotGEOmap(coastmap, border = "black", add = TRUE,
+     MAPcol = NA)
> vertical.image.legend(col=colormap,
+     zlim = range(ground$z[1,ind,,]) - 272.15)
> #Turn of PNG device
> #Omit this line if you want to display image
> #rather than save to file
> dev.off()
```

```
null device
          1

> #FIGURE 2
> #Temperature at 300 mb
> ind <- which(atmos$variables == "tmpprs")
> colormap <- rev(rainbow(100, start = 0 , end = 5/6))
> png(file = "fig_300mb_temp.png", width = 1000, height = 750)
> image(atmos$x, atmos$y, atmos$z[1,ind,,], col = colormap,
+     xlab = "Longitude", ylab = "Latitude",
+     main = paste("World Temperature at 300 mb:", atmos$fcst.date))
> plotGEOmap(coastmap, border = "black", add = TRUE,
+     MAPcol = NA)
> vertical.image.legend(col=colormap,
+     zlim = range(atmos$z[1,ind,,]) - 272.15)
> dev.off()

null device
          1

> #FIGURE 3
> #Relative humidity at ground level
>
> colormap <- rev(cm.colors(100))
> ind <- which(ground$variables == "rh2m")
> png(file = "fig_ground_rh.png", width = 1000, height = 750)
> image(ground$x, ground$y, ground$z[1,ind,,], col = colormap,
+     xlab = "Longitude", ylab = "Latitude",
+     main = paste("World Relative Humidity at Ground Level:",
+     ground$fcst.date))
> plotGEOmap(coastmap, border = "black", add = TRUE,
+     MAPcol = NA)
> vertical.image.legend(col=colormap,
+     zlim = range(ground$z[1,ind,,]))
> dev.off()

null device
          1
```

```
> #FIGURE 4
> #Winds at 300 mb (around jet stream level)
> ew.ind <- which(atmos$variables == "ugrdprs")
> ns.ind <- which(atmos$variables == "vgrdprs")
> winds.vel <- sqrt(atmos$z[1,ew.ind,,]^2 + atmos$z[1,ns.ind,,]^2)
> colormap <- terrain.colors(100)
> png(file = "fig_300mb_winds.png", width = 1000, height = 750)
> image(atmos$x, atmos$y, winds.vel, col = colormap,
+     xlab = "Longitude", ylab = "Latitude",
+     main = paste("World Wind Velocity at 300 mb:", atmos$fcst.date))
> plotGEOmap(coastmap, border = "black", add = TRUE,
+     MAPcol = NA)
> vertical.image.legend(col=colormap,
+     zlim = range(winds.vel))
> dev.off()

null device
          1

> #Get model data
> #and plot gust for 6 hr forecast
>
> variables <- c("gustsfc")
> time <- c(2,2) #6 hr forecast
> ground.wind.data <- DODSGrab(latest.model, latest.model.run,
+    variables, time, lon, lat, display.url = FALSE)
> #Make an array for quick indexing
> ground.wind <- ModelGrid(ground.wind.data, c(0.5, 0.5))
> #FIGURE 5
> png(file = "fig_ground_gust.png", width = 1000, height = 750)
> image(ground.wind$x, ground.wind$y, ground.wind$z[1,1,,], col = colormap,
+     xlab = "Longitude", ylab = "Latitude",
+     main = paste("World Wind Gust:",
+     ground.wind$fcst.date))
> plotGEOmap(coastmap, border = "black", add = TRUE,
+     MAPcol = NA)
> vertical.image.legend(col=colormap,
+     zlim = range(ground.wind$z[1,1,,]))
> dev.off()
```

```
null device
          1

>
```

2.2 Figures

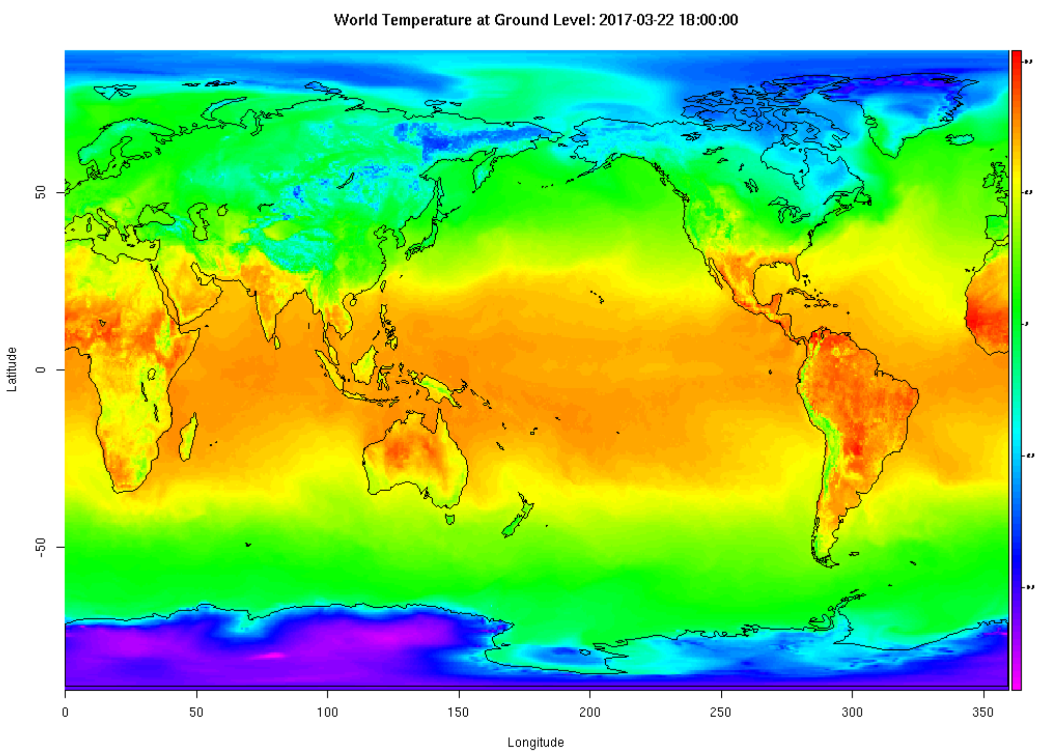

Figure 1: Temperature at 2 m above ground. Note how high altitude regions show up as cold spots.

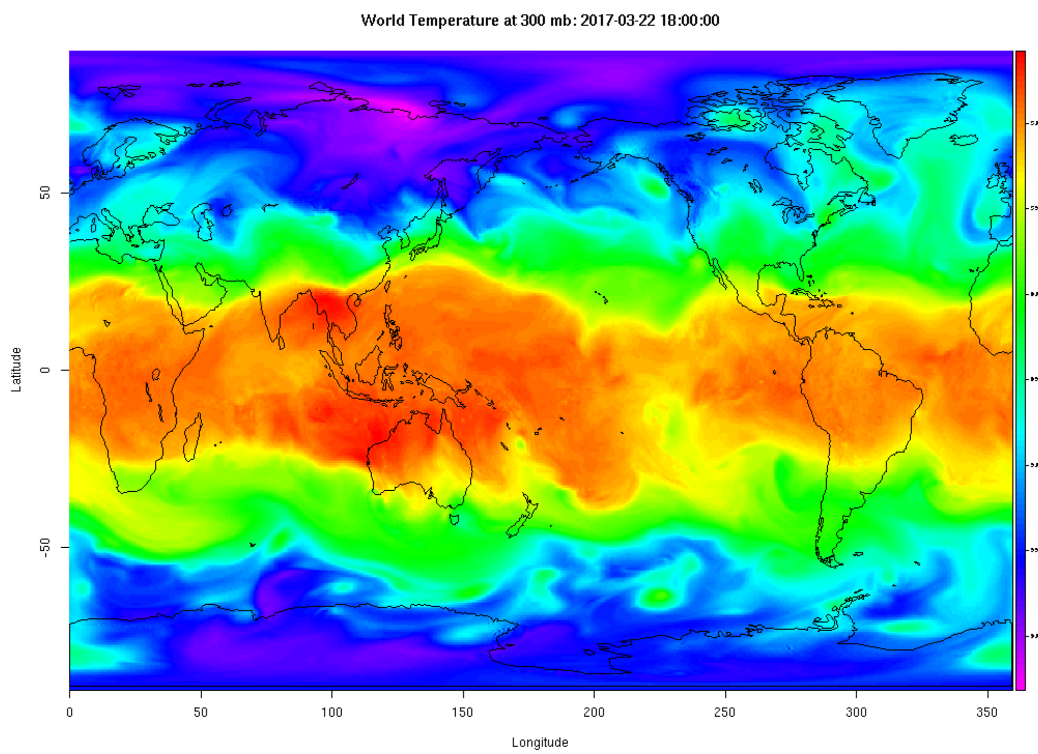

Figure 2: Temperature of the upper troposphere. Note how there is little relation between the location of continents and temperature.

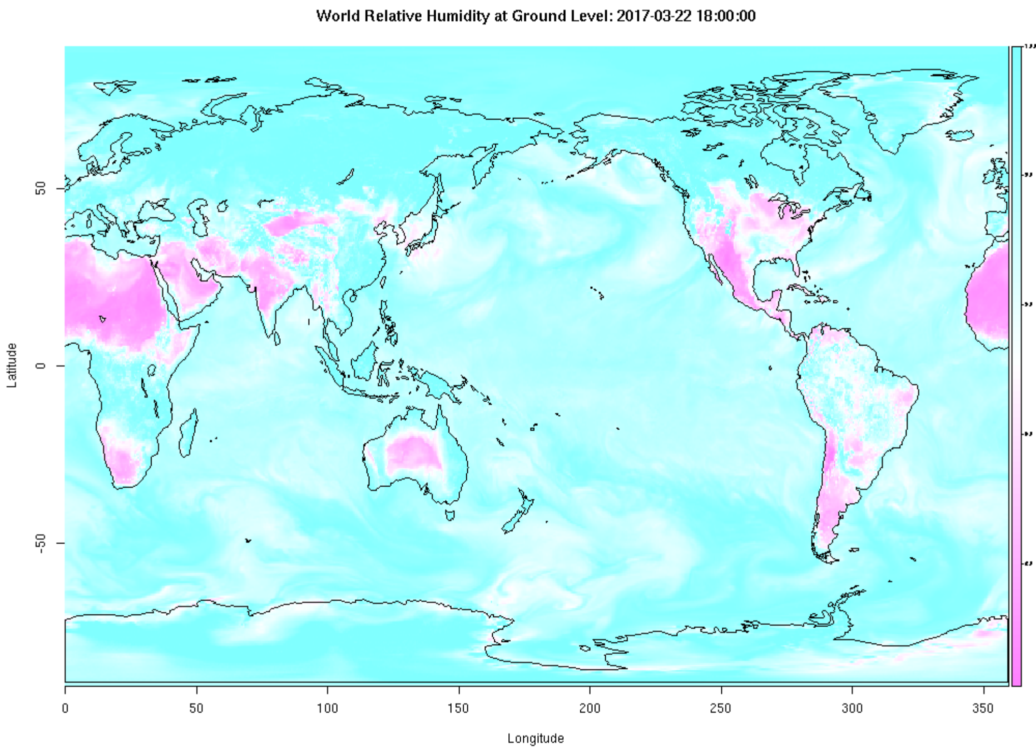

Figure 3: Relative humidity at 2 m above the ground. Desert regions show up nicely.

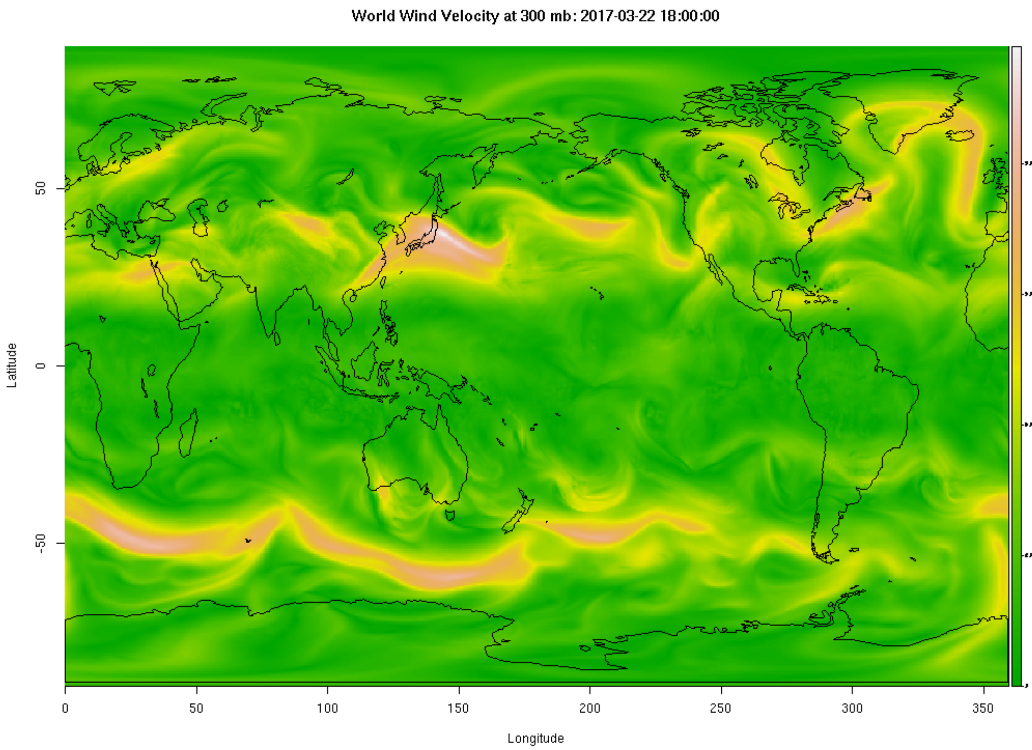

Figure 4: Wind velocity in the upper troposphere. Jet streams may show up here as patchy regions of intense wind.

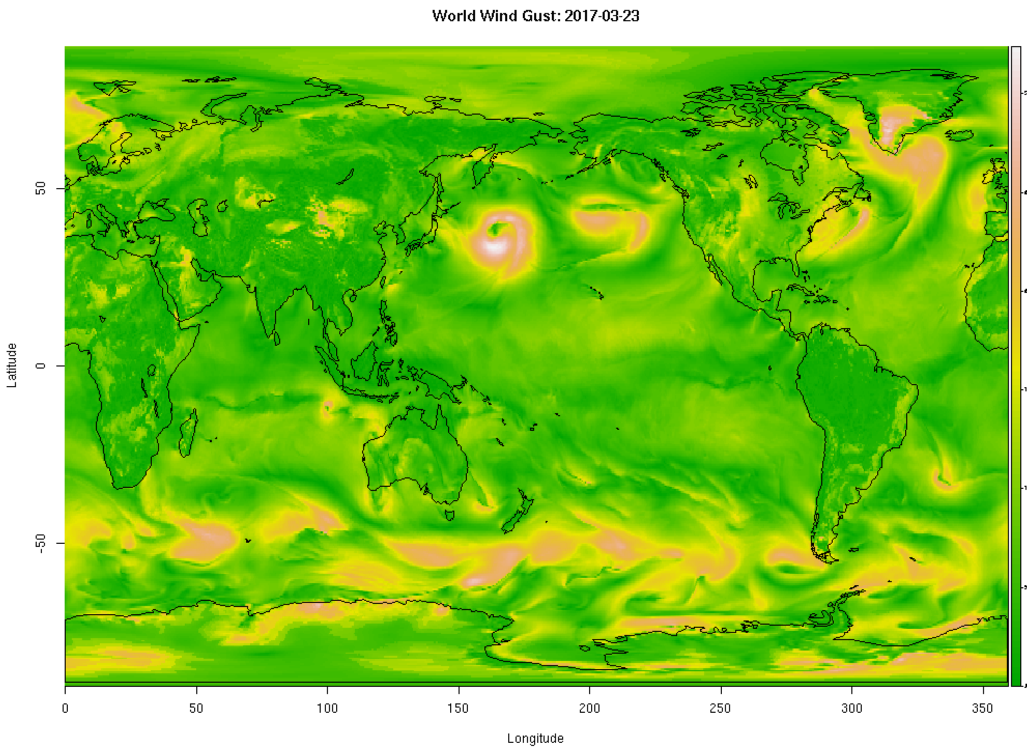

Figure 5: 6 hour prediction of wind gust speed at the ground surface. Hurricanes and tropical storms show up as small white dots.

3 Creating Atmospheric Profiles

We can generate data for specific points very quickly with rNOMADS. In this case, we will investigate the temperature and wind speed versus height directly above the University of North Carolina at Chapel Hill. The tropopause is clearly visible in the temperature profile (Figure 6). Wind speeds can be surprisingly high at times (Figure 7). We then show a quick calculation of sound velocity versus height (Figure 8) using the formula:

$$c = \sqrt{\gamma R_a T} \tag{1}$$

where c is the speed of sound, γ is the adiabatic index of air, R_a is the specific gas constant of air, and T is temperature. This does not take into account wind speed (a significant omission!).

Finally, we end with a calculation of atmospheric density versus height (Figure 9) via

$$\rho = \frac{p}{R_a T} \tag{2}$$

where ρ is density and p is pressure.

3.1 Code

```
> library(rNOMADS)
> #Location to examine
> lon <- -79.052104
> lat <- 35.907553
> #Figure out nearest model node to this point
>
> lons <- seq(0, 359.5, by = 0.5)
> lats <- seq(-90, 90, by = 0.5)
> lon.diff <- abs(lon + 360 - lons)
> lat.diff <- abs(lat - lats)
> model.lon.ind <- which(lon.diff == min(lon.diff)) - 1
> model.lat.ind <- which(lat.diff == min(lat.diff)) - 1
> model.urls <- GetDODSDates("gfs_0p50")
> latest.model <- tail(model.urls$url, 1)
> model.runs <- GetDODSModelRuns(latest.model)
> latest.model.run <- tail(model.runs$model.run, 1)
```

```
> #Get data
>
> time <- c(0,0) #Analysis model
> lev <- c(0, 46) #All levels in atmosphere
> variables <- c("tmpprs", "hgtprs", "ugrdprs", "vgrdprs")
> model.data <- DODSGrab(latest.model, latest.model.run,
+    variables, time, c(model.lon.ind - 2, model.lon.ind + 2),
+    c(model.lat.ind - 2, model.lat.ind + 2),
+    levels = lev, display.url = FALSE)
> #Build atmospheric profile
> profile <- BuildProfile(model.data, lon + 360, lat,
+     spatial.average = TRUE, points = 8)
> hgt <- profile[[1]]$profile.data[, which(variables == "hgtprs"),]
>

> #FIGURE 6 - temperature
>
>
> tmp <- profile[[1]]$profile.data[, which(variables == "tmpprs"),]
> #Let's make a spline
>
> tmp.spline <- splinefun(hgt, tmp, method = "natural")
> synth.hgt <- seq(min(hgt), max(hgt), length.out = 1000)
> synth.tmp <- tmp.spline(synth.hgt)
> png(file = "fig_tmp_profile.png", width = 500, height = 500)
> plot(tmp - 272.15, hgt, pch = 19, col = "red",
+    xlab = "Temperature (C)", ylab = "Height (m)",
+    main = paste("Temperature versus Geopotential Height:",
+    atmos$fcst.date))
> lines(synth.tmp - 272.15, synth.hgt, col = "blue")
> legend("topright", col = c("red", "blue"), pch = c(19, NA),
+    lty = c(NA, 1), legend = c("Model Values", "Spline Fit"),
+    bg = "white")
> dev.off()

null device
          1

>
```

```
> #FIGURE 7 - Wind Wheel
>
> meridional <- profile[[1]]$profile.data[, which(variables == "vgrdprs"),]
> zonal <- profile[[1]]$profile.data[, which(variables == "ugrdprs"),]
> #Let's make a spline
>
> merid.spline <- splinefun(hgt, meridional, method = "natural")
> zone.spline  <- splinefun(hgt, zonal, method = "natural")
> synth.hgt   <- seq(min(hgt), max(hgt), length.out = 1000)
> synth.merid <- merid.spline(synth.hgt)
> synth.zone  <- zone.spline(synth.hgt)
> png(file = "fig_wind_profile.png", width = 500, height = 500)
> PlotWindProfile(synth.zone, synth.merid, synth.hgt, lines = TRUE,
+     points = FALSE, elev.circles = c(0, 25000, 50000),
+     elev.labels = c(0, 25, 50),
+     radial.lines = seq(45, 360, by = 45), colorbar = TRUE, invert = FALSE,
+     point.cex = 2, pch = 19, lty = 1, lwd = 3,
+     height.range = c(0, 50000), colorbar.label = "Wind Speed (m/s)")
> dev.off()

null device
          1

>

> #FIGURE 8
> #NAIVE SOUND SPEED
> #Assuming it only depends on temperature
>
> R.sp  <- 287.058 #Specific gas constant of air
> gamma <- 1.4 #Adiabatic index
> tmp.spline <- splinefun(hgt, tmp, method = "natural")
> synth.hgt <- seq(min(hgt), max(hgt), length.out = 1000)
> synth.tmp <- tmp.spline(synth.hgt)
> c       <- sqrt(gamma * R.sp * tmp)
> synth.c <- sqrt(gamma * R.sp * synth.tmp)
> png(file = "fig_sound_profile.png", width = 500, height = 500)
> plot(c, hgt, pch = 19, col = "red",
```

```
+    xlab = "Speed of Sound (m/s)", ylab = "Height (m)",
+    main = paste("Speed of Sound versus Geopotential Height:",
+    atmos$fcst.date))
> lines(synth.c, synth.hgt, col = "blue")
> legend("topright", col = c("red", "blue"), pch = c(19, NA),
+    lty = c(NA, 1), legend = c("Model Values", "Spline Fit"),
+    bg = "white")
> dev.off()

null device
          1

>

> #FIGURE 8
> #ATMOSPHERIC DENSITY (ASSUMING DRY AIR)
>
> p <- unique(model.data$levels) * 100
> rho <- p / (tmp * R.sp) #Air density
> rho.spline <- splinefun(hgt, rho, method = "natural")
> synth.hgt <- seq(min(hgt), max(hgt), length.out = 1000)
> synth.rho <- rho.spline(synth.hgt)
> png(file = "fig_density_profile.png", width = 500, height = 500)
> plot(rho, hgt, pch = 19, col = "red",
+    xlab = "Density (kg/m3)", ylab = "Height (m)",
+    main = paste("Dry Air Density versus Geopotential Height:",
+    atmos$fcst.date))
> lines(synth.rho, synth.hgt, col = "blue")
> legend("topright", col = c("red", "blue"), pch = c(19, NA),
+    lty = c(NA, 1), legend = c("Model Values", "Spline Fit"))
> dev.off()

null device
          1

>
```

3.2 Figures

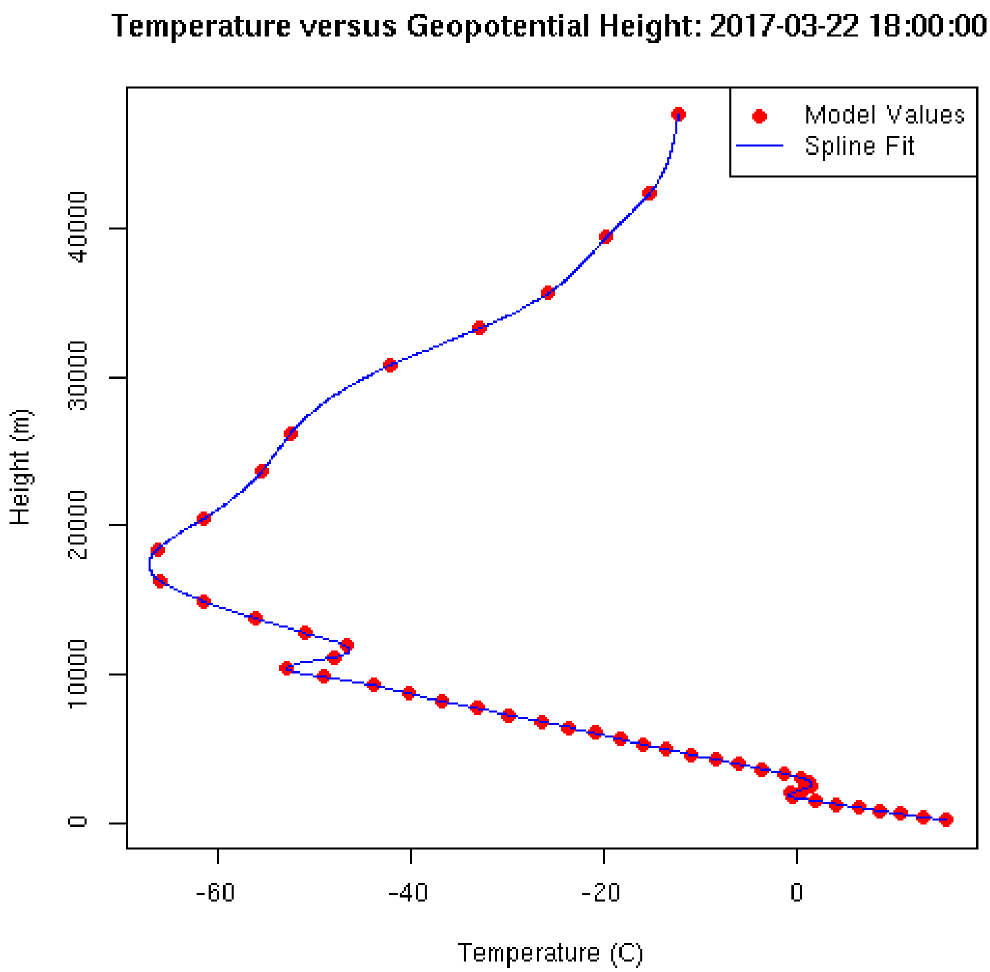

Figure 6: Temperature profile of the atmosphere near Chapel Hill, North Carolina.

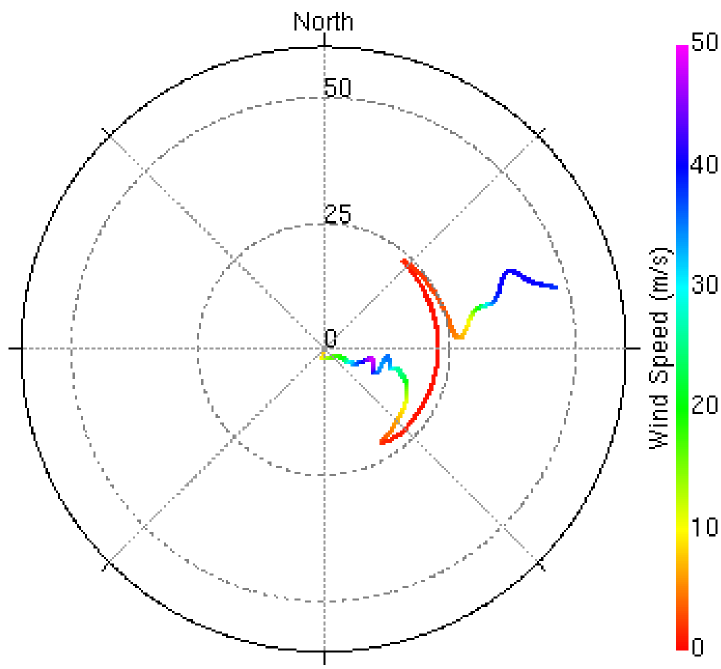

Figure 7: Wind profile of the atmosphere near Chapel Hill, North Carolina.

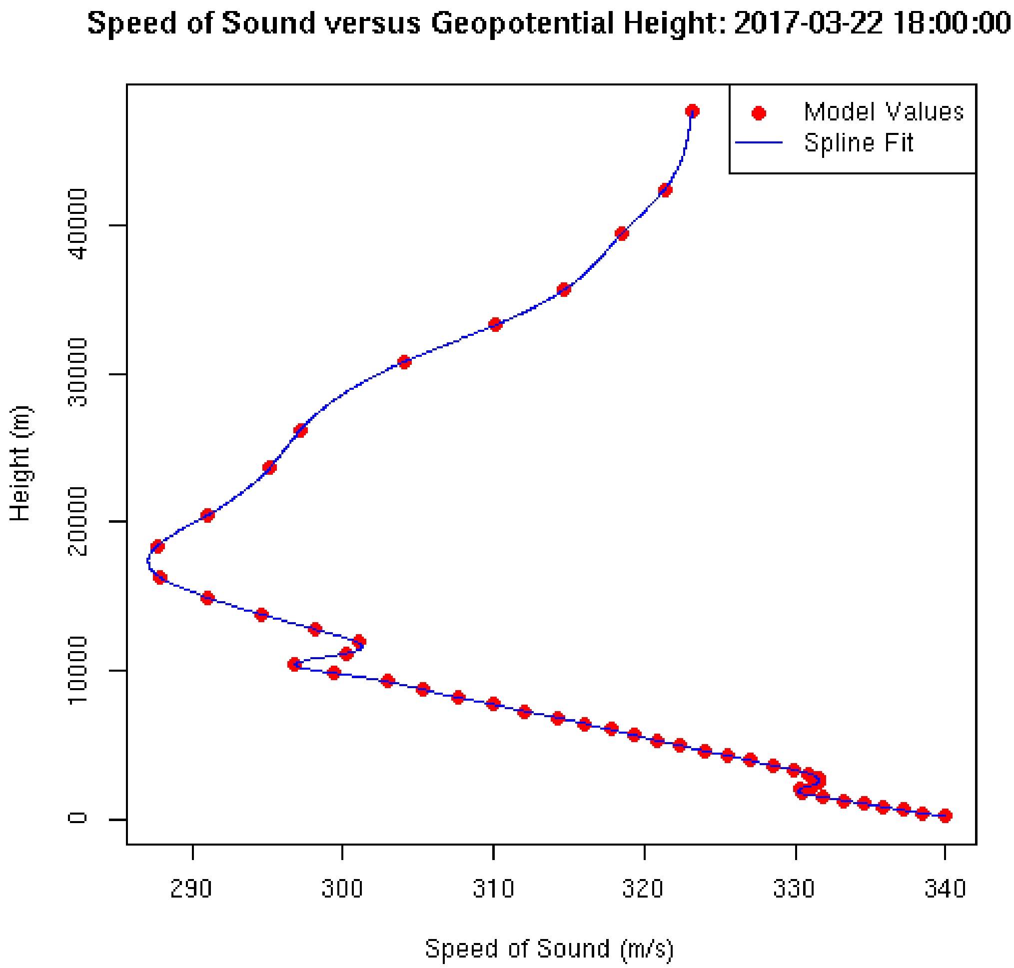

Figure 8: Sound speed profile of the atmosphere near Chapel Hill, North Carolina. This calculation assumes temperature dependence only.

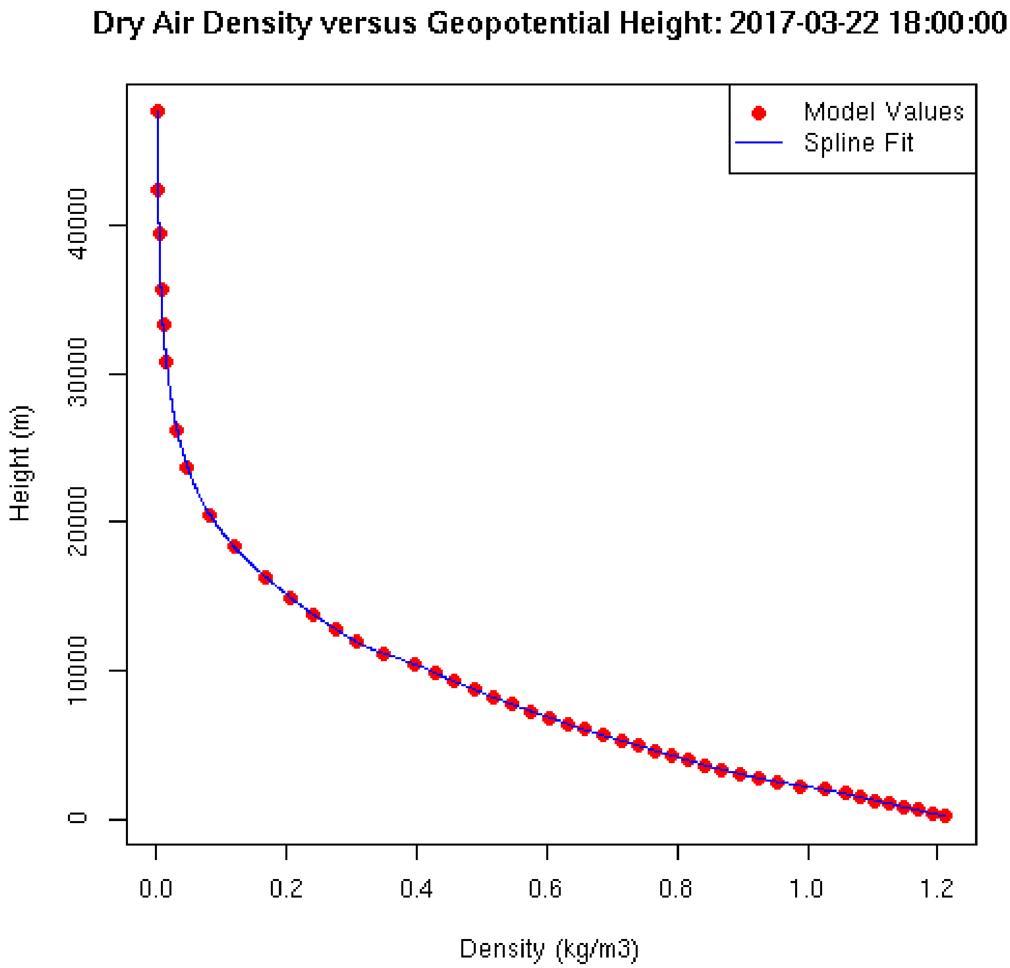

Figure 9: Density profile of the atmosphere near Chapel Hill, North Carolina.