Users' Guide to the NCAR Gust Front Tool V1.0: A System to Identify and Visualize Gust Fronts from Mesoscale Model Output

Developed by:
James F. Bresch, Craig S. Schwartz, Jordan G. Powers,
Ryan A. Sobash, and Janice L. Coen

December 2019

Table of Contents

1. Introduction

With the development and availability convective-permitting mesoscale numerical weather prediction (NWP) models that are capable of resolving circulations associated with convective systems, it is now possible to use the output from such systems to identify impactful convective features and effects. One class of features that can bear strongly on firefighter safety and incident operations is that of gust fronts, and thus their detection in the vicinity of wildland fires can be of critical value. Given this, and to aid fire weather forecasters, a tool has been developed to objectively identify and plot the locations of gust fronts in model output. The tool has various capabilities to plot the boundaries depending on the needs of forecasters and the type of model input. The tool can plot several model fields from a single, deterministic model run that have been traditionally used to identify gust front locations. In addition, the tool can process output from a forecast ensemble, produce wildfire-relevant products in formats commonly available from ensemble systems, such as paintball plots, postage stamp plots, and probability plots.

The tool was developed and tested using output from NOAA's High-Resolution Rapid Refresh (HRRR) system, a 3-km deterministic forecasting system, and the NCAR mesoscale ensemble, which also employs a 3-km grid. Both of those model systems use the Advanced Research Weather Research and Forecasting (WRF) Model, although the tool should work with any model's 3-km output. The NCAR ensemble was run for the years 2015 to 2017, and its output has been stored by NCAR for research purposes. The HRRR is available in real-time from servers of the National Centers for Environmental Prediction (NCEP).

The tool has been designed for the Unix operating system and requires a Fortran compiler, the Python scripting language, and commonly-available Python libraries. Sufficient disk space is also needed to store the model forecast output files.

Questions about the tool and its implementation may be sent to Dr. Jim Bresch (bresch@ucar.edu).

2. Configuring the Tool and its Environment

The following describes the steps for obtaining and installing the various components needed for the tool software.

First, the current software package can be downloaded from:
https://www2.mmm.ucar.edu/people/bresch/jfsp/ncar_gft.tar.gz .

After downloading to a Linux system, change to the directory where you wish to install the code, and untar the file:

tar xvf ncar_gft.tar.gz .

The code contains Fortran elements, and the Makefile to build the code may need editing to use the correct Fortran compiler. The default compiler is Intel (ifort). Before building the code, first set the environment variables appropriate for your model output.

GRIBPATH – This is the complete path to the GRIB libraries which are needed for HRRR output.

Ex: setenv GRIBPATH /hostdir/griblibs

NETCDFPATH – This is the complete path to the NetCDF libraries, which are needed for processing NetCDF format model output, such as that of WRF as run in the NCAR ensemble.

Ex: setenv NETCDFPATH /hostdir/netcdflibs

After this preparation, then simply type 'make'. This will create the gust front tool executable 'gf_tool.x'.

To remove the executable and object files (e.g., to re-build the code using a different compiler), type 'make clean'.

Use the same compiler to create all libraries and executables. Make sure they all specify the same bit size (typically 64-bit).

As referenced above, supporting libraries are needed to compile the tool. These are GRIB and NetCDF. If NetCDF is not already on your system it can be obtained from:

https://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html .

NetCDF Fortran libraries are needed for the tool, and they must be built after the C libraries have been created. Detailed instructions are provided on the Unidata site, and the Fortran build is described here:

https://www.unidata.ucar.edu/software/netcdf/docs/building_netcdf_fortran.html .

The GRIB libraries are created by installing NCEP's Universal Post Processor (UPP) software. Information on the UPP can be found here: https://dtcenter.org/upp-online-tutorial-uppv4-0/installing-upp .

As described in the UPP documentation, the UPP depends on NCEP's GRIB software. The GRIB version to use is GRIB2, and it can be downloaded from: https://www.nco.ncep.noaa.gov/pmb/codes/GRIB2/ .

3. Obtaining Model Output

HRRR data is available in real-time from NCEP's model servers. The format of the site URL is as follows:

ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/hrrr/prod/hrrr.ccyymmdd/conus/ .

Here *ccyymmdd* is the current 2-digit century, year, month, and day (e.g., 20200101). Note that NCEP retains only two days of forecasts on the server.

The relevant files are named: *hrrr.thhz.wrfprsfnn.grib2*, where *hh* in the 2-digit initial hour and *nn* is the 2-digit forecast hour. Forecasts are initialized hourly, and the output interval is also hourly. The forecast length is 18 hours except for the 00, 06, 12, and 18 UTC initializations, which run to 36 hours. Each file is approximately 400 MB, so a typical 18-h HRRR run requires nearly 8 GB of disk storage.

NCEP provides an interface for users to download a subset of the model output files at https://nomads.ncep.noaa.gov. By using the GRIB filter interface, it is possible for users to download only the levels, variables, and regions of interest. This would greatly cut down on the file sizes to be stored and the transfer times for acquiring them. While the tool does not support plotting subregions of the HRRR, downloading only the 2-D output fields would result in significant savings in transfer times and disk space.

NCAR ensemble output is available for the period 2015–to 2017 on the CISL (Computer and Information Systems Laboratory) community filesystem resource called *glade*. Model forecasts were run once per day starting at 0000 UTC with a forecast length of 48 hours. The WRF 3-km grid was 1580 x 985 points, and thus is substantial. To access this data from an NCAR computer simply set the file_top_dir name and the desired initialization times in the script *run_gf_tool.csh*.

4. Tool Operation

The tool consists of two main components. The first is the Fortran program 'gf_tool' which reads model output files, identifies gust front locations (and other relevant meteorological fields), and writes its output to a NetCDF file. The second component is a graphics script which plots user-requested fields from that NetCDF file. Two plotting scripts are provided: the primary, one written in Python, *make_python_plots.csh* and another written in NCL (NCAR Command Language), *plot_gf.ncl*. Users may also use their own plotting script or some other viewer (such as the NCL utility *ncview*) to visualize the gust front tool output.

a. Running the Fortran Program

A *csh* script has been provided in the tool's main directory to simplify running the Fortran program. *run_gf_tool.csh* is a simple script which can process multiple ensemble members, multiple initialization times, and multiple forecast hours. The shell variables necessary for running the script are located near the top of the file and are documented therein.

When the script is run, it overwrites the namelist file (*input.nml*) with the values specified. Namelists are a special type of Fortran input structure that is used to set variable values. The following is an example file (*input.nml*) which is created by the script to set values in the namelist denoted 'share'.

```
&share
ens_size=10,
input_file_list='./fnames_ensemble.txt',
input_file_list_prev_hour='./fnames_ensemble_prev.txt',
output_file_list='./fnames_output.txt'
fname_grid_info='./rt_ensemble_wrfinput_d02',
search_radius_precip=40.,     ! distance (km) within which
to search for precip >= precip_thresh
precip_thresh = 0.10
search_radius_refl=40.,       ! distance (km) within which
to search for column-max reflectivity >= refl_thresh
```

```
refl_thresh = 10.0,
wind_diff_min_mag = 3.0,
wind_diff_min_ang = 10.0,
gaussian_length_scale=6
nms_window_size=10
num_binary_dilations=0
min_gf_size=10
&end
```

The following table provides definitions and values for the tool's namelist variables. The gust front tool can be tuned by adjusting these values. Caution is advised when making changes to the values, however, as results can be sensitive to the settings.

Namelist variable	Description	Default	Recommended value
ens_size	number of ensemble members (10 or fewer for NCAR ens., 1 for HRRR)	1	10
input_file_list	List of model file names created by run_gf_tool.csh		
input_file_list_prev_hour	created by run_gf_tool.csh		
output_file_list	created by fnames_output.txt		
fname_grid_info	file which contains grid definition (e.g. wrfinput). Set in run_gf_tool.csh		
search_radius_precip	radius (km) within which to search for precip >= precip_thresh (-1 means disregard, don't use 0)	0	40.
precip_thresh	minimum precipitation (mm) to search for in the neighborhood (search_radius_precip) to define event	0	0.10
search_radius_refl	distance (km) within which to search for column-max reflectivity >= refl_thresh (-1 means disregard, don't use 0)	0	40.
refl_thresh	minimum col-max reflectivity to search for in the neighborhood (search_radius_refl) to define event	0	10.0
wind_diff_min_mag	minimum threshold (m/s) of 1-h vector wind difference required for a gust front to be present	0	3.0
wind_diff_min_ang	minimum angle (degrees) of 1-h wind direction shift	0	10.0

	required for a gust front to be present		
gaussian_length_scale	gaussian_length_scale	0	6
nms_window_size	window size for NMS. an integer that is the halfwidth of rectangle. total number of points is 2*nms_window_size. Set to < 0 to deactivate NMS	0	10
num_binary_dilations	number of binary dilations applied to NMS. only active if nms_window_size > 0	0	0
min_gf_size	minimum size of the gust front (number of contiguous pixels)	0	10

b. Gust Front Tool Output File

When the gust front program finishes successfully, it creates a NetCDF output file of the form *gf_tool_memx_ccyymmddhh_fttt.nc*, where *x* is the member number, *ccyymmddhh* is the initial time of the model run (century, year, month, day, hour in UTC), and *ttt* is the 3-digit forecast hour. The file contains several 2-d fields which may be plotted to aid in identification of gust fronts. The penultimate field is named ‘binary_gust_front’ and contains the objectively-determined gust front locations.

Variable Name	Description
TH2	2-m potential temperature (K)
U10	10-m zonal wind (m/s)
V10	10-m meridional wind (m/s)
tot_precip	Total accumulated precipitation (mm)
REFL_MAX	Column-max reflectivity (dBZ)
grad_theta	Gradient of potential temperature (K/100km)
FRONTdiv	Frontogenesis divergence term (K/100km/h)
FRONTdef	Frontogenesis deformation term (K/100km/h)
FRONTtot	Total frontogenesis (K/100km/h)
GradFRONTtot	Magnitude of gradient of total frontogenesis (K/100km^2)/h)
GradFRONTtotAng	Angle of the gradient of total frontogenesis (degrees)
VEC_WIND_DIFF_MAG	1-hr vector wind difference magnitude (m/s)
VEC_WIND_DIFF_ANGLE	1-hr vector wind difference angle (degrees)
binary_gust_front	Gust front (0 or 1)

c. Modifying the Gust Front Program

Tests of the gust front tool showed that objectively identifying gust fronts using the surface potential temperature (θ) gradient as the target meteorological field produced the best results (e.g., correct positioning, fewest false alarms). However, it is possible for users to employ different target fields (for example, to use frontogenesis instead of potential temperature gradient) by modifying the source code. Users could then perform their own research to better understand the behavior or limitations of the tool. For this, simply edit the source code files as desired and type 'make' to rebuild the code.

5. Running the Plotting Capability

The main script to create gust front plots is called *make_python_plots.csh*. The script relies on three Python files (each with a suffix of .py) in the main software directory—fieldinfo.py, make_webplot.py, and webplot.py. It is important to note that at this time, the system only runs using Python 2.7; the programs will not work with Python 3.

For users with access to the NCAR community compute resource, on that platform first type the following commands.

module load python/2.7.16

module load ncarenv

ncar_pylib

These load the Python packages, and the necessary Python libraries are listed at the top of the webplot.py program. Also needed by the script is the *colormap*s directory, the *cmap_rad.rgb* file and the *latlon.nc* file.

The *make_python_plots.csh* script creates a set of plots from the tool output. At the top of the script are several shell variables that can be set to control the start time, forecast hour, forecast interval, and the areas of the plots. These are shown in the table below. The script determines the file name of the gust front tool output file (i.e., the NetCDF file produced by the Fortran program) from the initial time and forecast hour.

csh variable	Description	Example value
start_init	10-digit initial time of the form *ccyymmddhh*.	2017072500
final_init	10-digit final time to process. Set to start_init to process one forecast.	2017072500
inc_init	Increment between initial times in hours.	24 (for the NCAR ensemble)

fhrs	Forecast hours to plot. Can be a list of hours or use the unix seq command as shown in the example.	(`seq 21 1 27`) or (21 22 23 24)
storage_top	Directory where output plots are placed (typically set to $PWD).	$PWD
workdir	Directory where the script is run (typically set to $PWD). Temporary files are placed here and deleted.	$PWD
center_lat	Center latitude for the plot. Decimal degrees with northern hemisphere positive.	42.28
center_lon	Center longitude for the plot. Decimal degrees with western hemisphere negative	-110.95
ENS_SIZE	Ensemble size. 10 for NCAR ensemble; 1 for the HRRR.	10

The plot area is determined by the *center_lat* and *center_lon* shell variables and the variables 'deltax' and 'deltay' located in the webplot.py program. The default value for these variables is 4.0 degrees latitude, although this can be changed by editing *webplot.py*.

Users can control the plots created by the script by editing *make_python_plots.csh* to comment or uncomment the calls to *make_webplot.py*. By default, the script produces a neighborhood probability plot, a binary gust front paintball plot, a composite reflectivity postage stamp plot, and a precipitation postage stamp plot. These plots are also produced for HRRR output; however, until NCEP runs an operational HRRR ensemble, the probability plots should be interpreted carefully. With one member, the HRRR probability plot only has a value of 0 or 1, simply indicating whether or not a gust front is predicted within 40 km of a given point by the one forecast.

6. Examples of Tool Output Graphics

Sample output produced by the tool are presented below. These illustrate probability plots, paintball plots, and postage stamp plots.

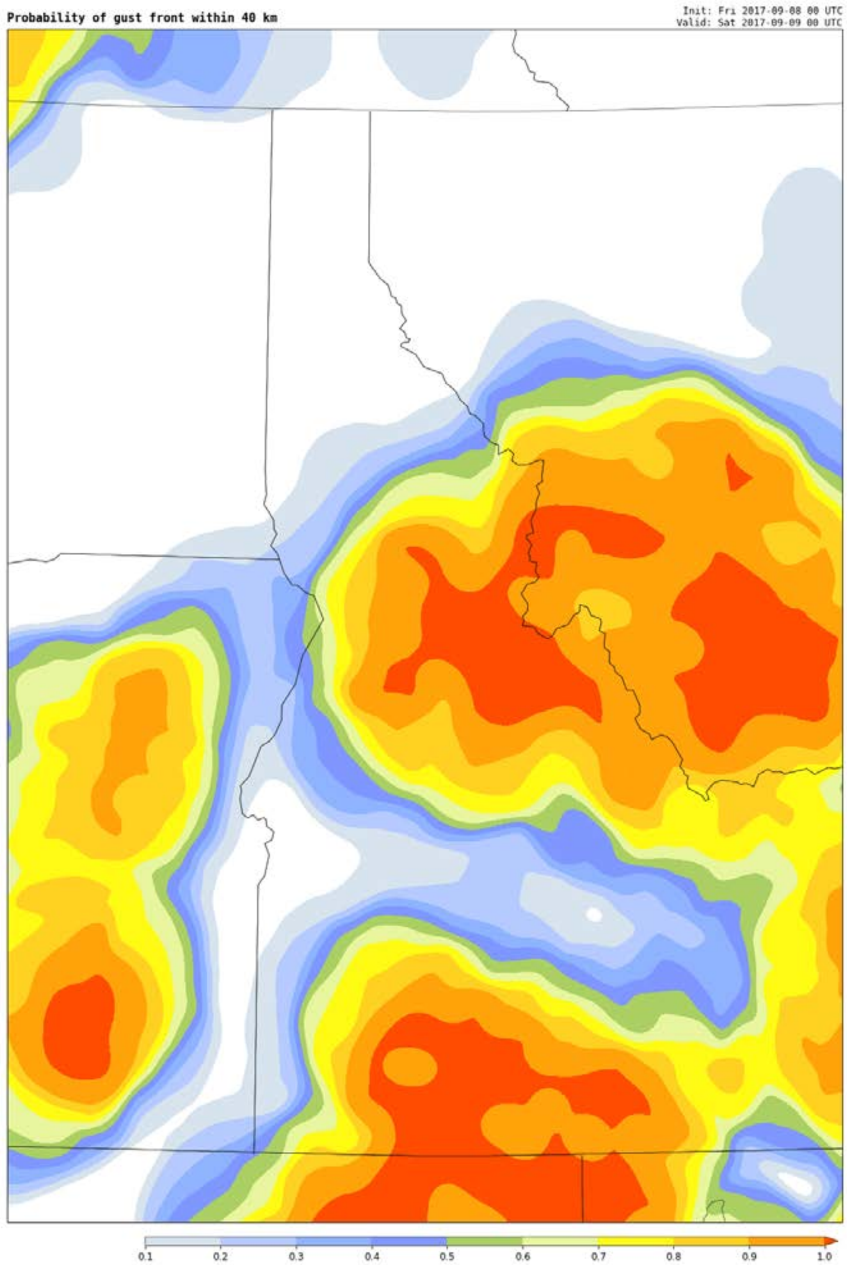

Figure 1: Probability of gust fronts within 40 km of a given point for the Idaho region. This reflects 24-h forecasts from the 10-member NCAR ensemble, valid at 0000 UTC 9 September 2017. Red shading means 100% of the members forecasted a gust front, white= 0%. Probability scale at bottom..

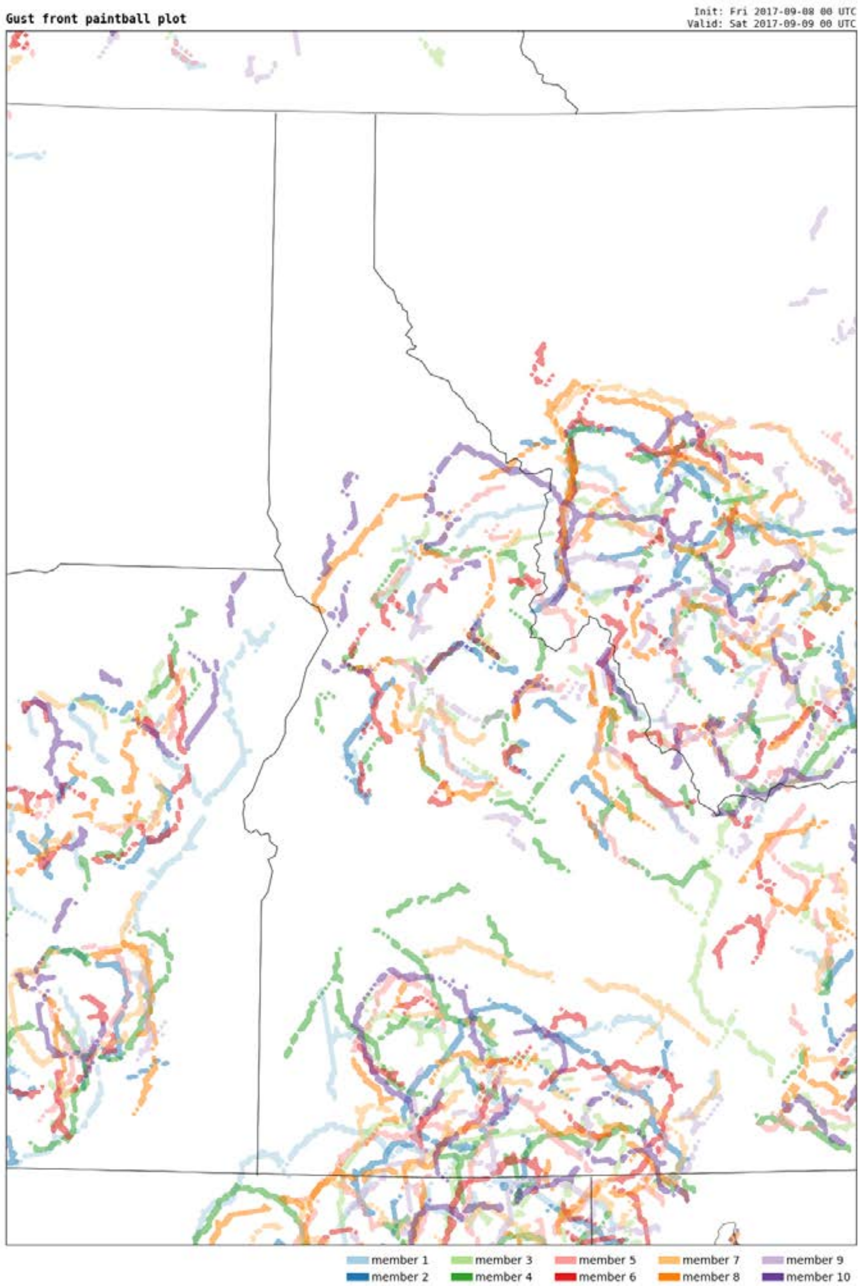

Figure 2: As in Figure 1, but displaying objectively-determined gust front boundaries . Each member's forecast is colored according to the legend at bottom.

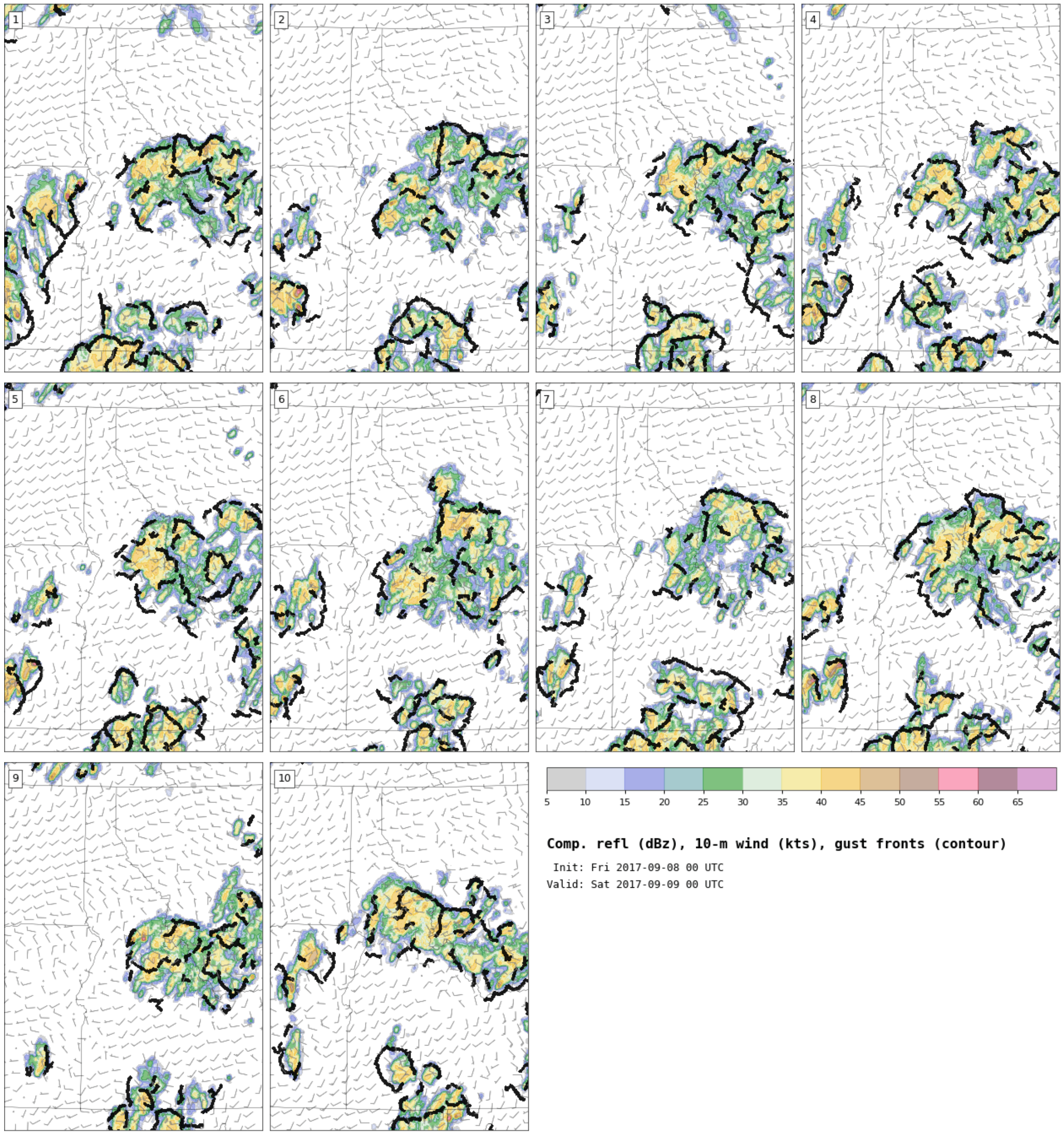

Figure 3: As in Figure 2, but showing composite reflectivity (dBZ, shaded), gust front locations (black line segments) and 10-m wind barbs (full barb= 5 m/s).

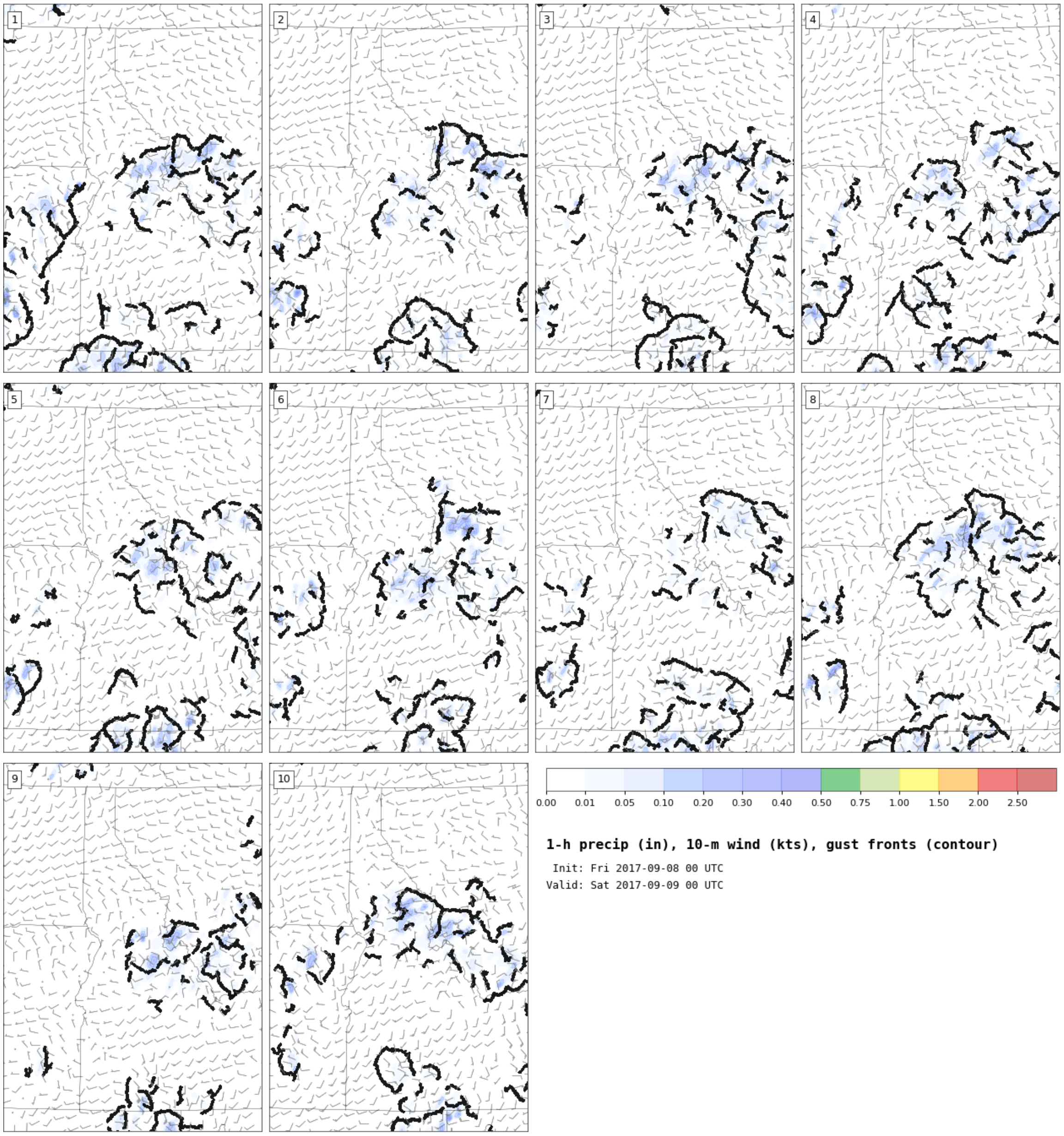

Figure 4: As in Figure 3, but showing identified gust fronts overlaid on model-forecast 1-h precipitation (inches, color shaded).

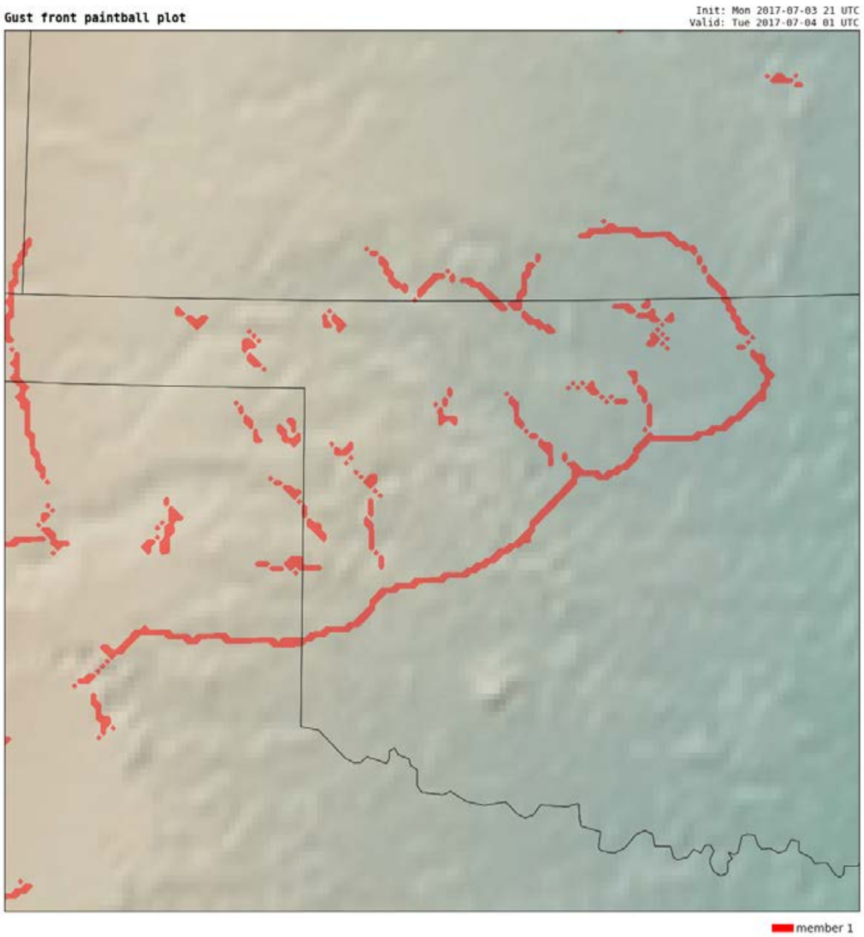

Figure 5: As in Figure 2, but for objectively-determined gust front boundaries in red for a single 4-h HRRR forecast valid at 0100 UTC 4 July 2017 over Oklahoma and Texas. Terrain appears as a shaded background field.

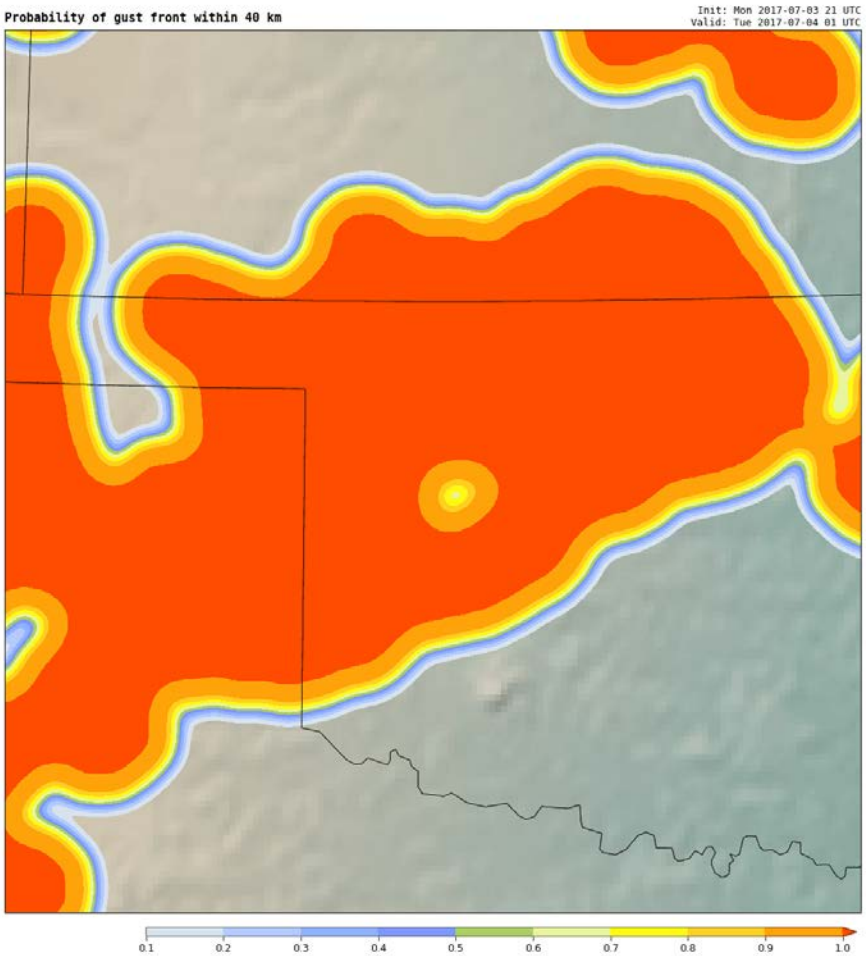

Figure 6: As in Figure 1, but for the single HRRR forecast shown in Figure 5. Red shading shows where the HRRR forecast predicts a gust front within 40 km of a given point. This is a binary field (i.e., actual probabilities from the single forecast are either 0 or 1, but interpolation causes the color shading on the perimeter of the 100% area.