Stochastic Time-Inverted Lagrangian Transport (STILT) Model

www.stilt-model.org

How To

by

John C. Lin
jcl@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Christoph Gerbig
christoph.gerbig@bgc-jena.mpg.de
Max-Planck-Institut für Biogeochemie
Jena, Germany

August, 2010

Max-Planck-Institut
für Biogeochemie

UNIVERSITY OF
Waterloo

aer
Atmospheric and
Environmental Research, Inc.

Table of Contents

Foreword

The Stochastic Time-Inverted Lagrangian Transport (STILT) model is an atmospheric model that simulates the movement of air parcels using ensembles of fictitious "particles" starting from a specified location and time. These particles can run both backward and forward in time from the starting location. STILT has predominantly been used in a time-reversed manner, where the particles spread out as they travel backward in time. In this way, the particles provide knowledge of the trajectories of air that arrives at the specified location (receptor). From this information one can construct the "influence" or "footprint" of an atmospheric observation and has proven to be extremely useful for understanding atmospheric datasets. As a simple example, imagine a case in which one observes an increase in pollutant concentrations. By examining the STILT-simulated particles, then, the researcher can get a sense of where the plume originated from.

This document is intended to provide a quick introduction to running STILT and serves as a supplement to the official STILT website: http://www.stilt-model.org/

Fair Use Policy

STILT is freely available and we encourage others to use it. Kindly keep us informed of how you are using the model and of any publication plans. Please acknowledge the source as a citation. STILT is continuously updated and improved by the development consortium, and in some cases (as when new elements are used for the first time) we may suggest that one or more of the developers be offered participation as authors. If your work directly competes with our analysis we may ask that we have the opportunity to submit a manuscript before you submit one that uses unpublished features. The software is updated from time to time, and it is your responsibility to insure that your publication is consistent with the most recent revision.

Publications

The formal reference describing the STILT model is:

Lin, J.C., C. Gerbig, S.C. Wofsy, A.E. Andrews, B.C. Daube, K.J. Davis, and C.A. Grainger, A near-field tool for simulating the upstream influence of atmospheric observations: the Stochastic Time-Inverted Lagrangian Transport (STILT) model, *Journal of Geophysical Research*, *108* (D16), 4493, doi:10.1029/2002JD003161, 2003.

A few selected papers that use STILT:

Gerbig, C., J.C. Lin, S.C. Wofsy, B.C. Daube, A.E. Andrews, B.B. Stephens, P.S. Bakwin, and C.A. Grainger, Toward constraining regional-scale fluxes of CO_2 with atmospheric observations over a continent: 2. Analysis of COBRA data using a receptor-oriented framework, *Journal of Geophysical Research*, *108* (D24), 4757, doi:10.1029/2003JD003770, 2003.

Lin, J.C., et al., Measuring fluxes of trace gases at regional scales by Lagrangian observations: application to the CO_2 Budget and Rectification Airborne (COBRA) study, *Journal of Geophysical Research*, *109* (D15304, doi:10.1029/2004JD004754), 2004.

Lin, J.C., and C. Gerbig, Accounting for the effect of transport errors on tracer inversions, *Geophysical Research Letters*, *32* (L01802), doi:10.1029/2004GL021127, 2005.

Hurst, D.F., J.C. Lin, P.A. Romashkin, B.C. Daube, C. Gerbig, D.M. Matross, S.C. Wofsy, B.D. Hall, and J.W. Elkins, Continuing global significance of emissions of Montreal Protocol-restricted halocarbons in the United States and Canada. *Journal of Geophysical Research*, *111(D15): D15302*, DOI: 10.1029/2005JD006785, 2006.

Kort, E.A., J. Eluszkiewicz, B.B. Stephens, J.B. Miller, C. Gerbig, T. Nehrkorn, B.C. Daube, J.O. Kaplan, S. Houweling, and S.C. Wofsy, Emissions of CH_4 and N_2O over the United States and Canada based on a receptor-oriented modeling framework and COBRA-NA atmospheric observations. *Geophysical Research Letters*, *35*: L18808, doi:10.1029/2008GL034031, 2008.

Macatangay, R., T. Warneke, C. Gerbig, S. Körner, R. Ahmadov, M. Heimann, and J. Notholt, A framework for comparing remotely sensed and in-situ CO_2 concentrations. *Atmospheric Chemistry and Physics, 8*: 2555-2568, 2008.

Miller, S.M., D.M. Matross, A.E. Andrews, D.B. Millet, M. Longo, E.W. Gottlieb, A.I. Hirsch, C. Gerbig, J.C. Lin, et al., Sources of carbon monoxide and formaldehyde in North America determined from high-resolution atmospheric data, *Atmospheric Chemistry and Physics, 8*, 7673-7696, 2008.

Nehrkorn, T., J. Eluszkiewicz, S.C. Wofsy, J.C. Lin, C. Gerbig, M. Longo, and S. Freitas. Coupled Weather Research and Forecasting--Stochastic Time-Inverted Lagrangian Transport (WRF-STILT) Model, *Meteorology and Atmospheric Physics, 107*, 51-64, 2010.

A complete list of STILT publications can also be found on the STILT website.

Content of Distribution Package

`Exe`	multiple copies of the STILT executables
`Boundary`	tracer boundary conditions, valid for the middle of the Pacific
`Fluxes`	surface flux grids over North America
`Rsc`	R scripts to run STILT and the ROAM framework (Receptor-Oriented Atmospheric Model)—i.e., calculating tracer concentrations at the particle starting location ("receptor") by having the STILT particles pick up tracer fluxes and tracer boundary conditions (see section on 'R scripts')
`src`	source code for the STILT model

Model Files Needed

The current version of the STILT model has been compiled to run on Linux machines.

Some compilers that have successfully compiled STILT:

- Portland Group Compiler (*pgf90*)
- Intel Fortran Compiler (*ifort*)
- GNU Fortran Compiler (*gfortran*)
- G95 Fortran Compiler (*g95*)

a) Essential files:

i) `hymodelc` compiled model executable file
ii) `CONTROL` file containing the main control parameters for the model
iii) `SETUP.CFG` file containing the secondary control parameters

b) Optional files

`ASCDATA.CFG`—configuration file defining data structure of `LANDUSE.ASC` & `ROUGLEN.ASC`
`LANDUSE.ASC`—text file containing land-use data
`ROUGLEN.ASC`—text file containing roughness length data

Note: The files above must be placed in the same directory.

Meteorological Files Needed

Because STILT does not solve the atmospheric equations of motions, it requires meteorological datasets to drive the movement of the particles. Since STILT is built upon the HYSPLIT (HYbrid Single-Particle Lagrangian Integrated Trajectory) model developed by NOAA-ARL (Air Resources Laboratory), the input meteorological files required by STILT are in NOAA-ARL format.

Archived meteorological files are available for download from NOAA-ARL:
http://ready.arl.noaa.gov/archives.php

A webpage that discusses the NOAA-ARL data format, with conversion programs, is available at:
http://ready.arl.noaa.gov/HYSPLIT_data2arl.php

The relevant papers describing the HYSPLIT model and the format of the NOAA-ARL meteorological files are:

Draxler, R.R., HYSPLIT_4 User's Guide, *NOAA Technical Memorandum ERL ARL-230*, 1999.
Draxler, R.R., and G.D. Hess, Description of the HYSPLIT_4 modeling system, *NOAA Technical Memorandum ERL ARL-224*, 1997.
Draxler, R.R., and G.D. Hess, An overview of the HYSPLIT_4 modelling system for trajectories, dispersion, and deposition, *Australian Meteorological Magazine*, *47*, 295-308, 1998.

Please properly acknowledge NOAA ARL as appropriate and please be aware of limitations when publishing results using forecast meteorology from ARL. The following are instructions from http://ready.arl.noaa.gov/HYSPLIT_traj.php:

Publications using HYSPLIT results, maps or other READY products provided by NOAA ARL are requested to include an acknowledgement of, and citation to, the NOAA Air Resources Laboratory. Appropriate versions of the following are recommended:

Citation
Draxler, R.R. and Rolph, G.D., 2003. HYSPLIT (HYbrid Single-Particle Lagrangian Integrated Trajectory) Model access via NOAA ARL READY Website (http://www.arl.noaa.gov/ready/hysplit4.html). NOAA Air Resources Laboratory, Silver Spring, MD.

Rolph, G.D., 2003. Real-time Environmental Applications and Display sYstem (READY) Website (http://www.arl.noaa.gov/ready/hysplit4.html). NOAA Air Resources Laboratory, Silver Spring, MD.

Acknowledgment
The authors gratefully acknowledge the NOAA Air Resources Laboratory (ARL) for the provision of the HYSPLIT transport and dispersion model and/or READY website (http://www.arl.noaa.gov/ready.html) used in this publication.

Redistribution Permission
Permission to publish or redistribute HYSPLIT model results using forecast meteorological data from NOAA ARL can be obtained by providing relevant information (reason, to whom, from whom) via email to permission@www.arl.noaa.gov.

Structure of CONTROL file

Here is a sample CONTROL file (with explanation following # sign):

```
0 8 26 18           #yr ('0'=>2000), mon, day, hr [UT]
1                   #number of starting locations
45 -90.2 5          #starting location:  lat, lon, height [m AGL]
48                  #number of hours to run model (negative numbers denote backward run)
0                   #vertical motion option (0:data 1:isob 2:isen 3:dens 4:sigma)
10000.0             #top of model domain [m AGL]
1                   #number of input meteorological data
/users/jcl/Metdat/  #meteorological data directory
edas.subgrd.aug00.002      #meteorological filename
1
test
1
0.01
00 00 00 00 00
1
0.0 0.0
0.5 0.5
30.0 30.0
./
cdump
1
100
00 00 00 00 00
00 00 00 00 00
00 24 00
1
0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0
0.0
0.0
```

Structure of SETUP.CFG

SETUP.CFG is a 'namelist' file, which can alter the behavior of the model without having to recompile it.

Here is a sample SETUP.CFG file:

```
$SETUP
NUMPAR=50,
DELT=30,
TRATIO=0.75,
ISOT=0,
TLFRAC=0.1,
OUTFRAC=0.9,
NDUMP=1,
RANDOM=1,
OUTDT=0.0,
VEGHT=0.5,
NTURB=0,
ICONVECT=0,
WINDERRTF=0,
ZICONTROLTF=0,
IVMAX=11,
VARSIWANT='time','indx','lati','long','zagl','zsfc','samt','temp','dswf','mlht','dmas',
$END
```

Explanation of parameters in SETUP.CFG:

NUMPAR—number of particles to be run (default: 100)

DELT—integration timestep [min]; if set to 0.0, then timestep is dynamically determined

TRATIO—maximum fraction of gridcell to be traveled by a particle in a single integration timestep. This determines the timestep is DELT is set to be dynamic (default: 0.75)

ISOT—flag used to set the isotropic turbulence option. The default value of 0 results in the computation of horizontal turbulence from wind field deformation. Setting this flag to 1 results in the horizontal turbulence to be the same in both the U- and V- directions. (default: 0)

TLFRAC–the fraction of T_L (Lagrangian timescale) to set as timestep in dispersion subroutine. The smaller this fraction is, the more finely the turbulence is resolved. (default: 0.1)

NDUMP—is a flag that can be set to dump out all the particle/puff points at the end of a simulation to a file called PARDUMP. This file can be read at the start of a new simulation to continue the previous calculation. Valid NDUMP settings: 0 - no I/O, 1- read and write, 2 - read only, 3 - write only (default: 0)

RANDOM–flag that tells random number generator whether to have a different random sequence each time model is run (0-FALSE; 1-TRUE); if set FALSE, then generates same random sequence each time, which is useful for debugging purposes (default: 1)

OUTDT–interval to output data to PARTICLE.DAT [min]; 0.0 means output is written every timestep (default: 0.0)

NTURB–No Turbulence flag; 1 sets run to simulate mean trajectories, 0 sets run to include turbulence (default: 0)

VEGHT–height [fraction of PBL ht or m] below which a particle's time spent is tallied; useful if want to specify a certain ht as 'seeing' ground vegetation. If <=1.0, then specifies fraction of PBL ht (default: 0.5)

OUTFRAC—the fraction of particles that are allowed to leave model domain (given by met. data); if exceeded, model stops (default: 0.9)

`ICONVECT`—flag for convection. If set to 1, then runs excessive convection as described in Gerbig et al., *J. Geophys. Res.*, *108* (D24), 4757, doi:10.1029/2003JD003770, 2003. For specialized RAMS output, the particles will be vertically redistributed according to the outputted convective mass fluxes (default: 0)

`WINDERRTF`—flag that specifies whether to have particle motions be affected by horizontal wind errors. (default: 0) If set to 1, then STILT looks for file called "WINDERR" that has four lines, with one number on each line: 1. Standard deviation of errors [m/s] 2. Correlation timescale of errors [min] 3. Vertical correlation lengthscale of errors [m] 4. Horizontal correlation lengthscale of errors [km]. All the statistical properties specified in 1.~4. are applied equally to the U- and V- wind components.

`ZICONTROLTF`—flag that species whether to scale the PBL heights in STILT uniformly in the entire model domain. (default: 0) If set to 1, then STILT looks for file called 'ZICONTROL' that species the scaling for the PBL height. The first line of ZICONTROL indicates the number of hours that the PBL height will be changed, and each subsequent line indicates the scaling factor for that hour. A sample ZICONTROL file could contain:

```
2
1.5
0.4
```

This file would alter the PBL heights for the first two hours of the model runtime, increasing the PBL height by 50% during the first hour and decreasing it to only 40% of the original value.

`IVMAX`—the total number of variables to be outputted (see `VARSIWANT`) (default: 5)

`VARSIWANT`—a list of 4-letter codes specifying variables to be outputted. See next section for complete list of variables that can be outputted (default: 'time', 'indx', 'long', 'lati', 'zagl')

List of Variables that can be Outputted from STILT

The following are the 4-letter codes of variables that should be specified for `VARSIWANT` in `SETUP.CFG` to get the output.

The output from STILT will be written to a text file called `PARTICLE.DAT`.

`time`	time since start of simulation; negative if going backward in time [min]
`indx`	particle index
`long`	longitude position of particle [degrees]
`lati`	latitude position of particle [degrees]
`zagl`	vertical position of particle [m above ground level]
`sigw`	standard deviation of vertical velocity; measure of strength of vertical turbulence [m/s]
`tlgr`	Lagrangian decorrelation timescale [s]
`zsfc`	terrain height [m above sea level]
`icdx`	cloud index when using RAMS (Grell scheme) [`1=updraft,2=environment,3=downdraft`]
`temp`	air temperature at lowest model layer [K]
`samt`	amount of time particle spends below VEGHT (see section on `SETUP.CFG`) [min]
`foot`	'footprint', or sensitivity of mixing ratio to surface fluxes [ppm/(μmole/m^2/s)]
`shtf`	sensible heat flux [W/m^2]
`lhtf`	latent heat flux [W/m^2]
`tcld`	total cloud cover [%]
`dmas`	particle weight changes due to mass violation in wind fields [initial value = 1.0]
`dens`	air density [kg/m^3]
`rhfr`	relative humidity fraction [0~1.0]
`sphu`	specific humidity [g/g]
`solw`	soil moisture
`lcld`	low cloud cover [%]
`zloc`	limit of convection heights [m]
`dswf`	downward shortwave radiation [W/m^2]
`wout`	vertical mean wind [m/s]
`mlht`	mixed-layer height [m]
`rain`	total rainfall rate [m/min]
`crai`	convective rainfall rate [m/min]

Sample PARTICLE.DAT output

The order of columns will be matched to the VARSIWANT specified in SETUP.CFG

```
42.53572       -72.17200        30.00000
  -30.       1.    42.5282   -72.1839    60.4405   263.4937    0.0563537      1703.51
  -30.       2.    42.5271   -72.1848    32.8217   263.6271    0.0563549      1703.51
  -30.       3.    42.5288   -72.1901    88.2987   264.9537    0.0563575      1703.51
  -30.       4.    42.5304   -72.1971    69.1384   267.0585    0.0563600      1703.51
  -30.       5.    42.5275   -72.1902   139.3386   264.9070    0.0563553      1703.51
  -60.       1.    42.5299   -72.2076   224.0589   269.1796    0.0743809      1283.81
  -60.       2.    42.5245   -72.2083   117.5103   268.4974    0.0743784      1283.81
  -60.       3.    42.5260   -72.2052     0.1150   268.7074    0.0743854      1283.81
  -60.       4.    42.5279   -72.2153   133.3466   270.7165    0.0743937      1283.81
  -60.       5.    42.5251   -72.2099   139.8024   268.9242    0.0743850      1283.81
  -90.       1.    42.5384   -72.2313   403.4171   275.8311    0.0836150       923.35
  -90.       2.    42.5294   -72.2307    38.5641   274.2447    0.0836082       923.35
  -90.       3.    42.5224   -72.2096     3.7323   269.3789    0.0835918       923.35
  -90.       4.    42.5319   -72.2384   217.6264   276.3353    0.0836206       923.35
  -90.       5.    42.5278   -72.2313   130.4637   274.2950    0.0836099       923.35
 -120.       1.    42.5518   -72.2531   394.9881   283.0028    0.1021289       923.35
 -120.       2.    42.5403   -72.2517    74.2778   280.1912    0.1021117       923.35
 -120.       3.    42.5238   -72.2223    58.4240   272.0490    0.1020613       923.35
 -120.       4.    42.5407   -72.2566   163.0003   282.3101    0.1021277       923.35
 -120.       5.    42.5340   -72.2526   107.5163   279.9136    0.1021114       923.35
```

The order of columns will be matched to the VARSIWANT specified in SETUP.CFG. Here the variables are: `'time','indx','lati','long','zagl','zsfc','foot','mlht'`

The CONTROL and SETUP.CFG files that generated the output above are as follows:

CONTROL file:

```
2 8 12 18
1
42.53572 -72.172 30
-2
0
25000.0
2
/bluejay/stilt/Metdata/
edas.subgrd.julaug02
/bluejay/stilt/Metdata/
fnl.nh.julaug02
1
test
1
0.01
00 00 00 00 00
1
0.0 0.0
0.5 0.5
30.0 30.0
./
cdump
1
100
00 00 00 00 00
00 00 00 00 00
00 2 00
1
0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0
0.0
0.0
```

SETUP.CFG file:

```
&SETUP
 TRATIO=0.75,
 INITD=0,
 KHMAX=9999,
 NUMPAR=5,
 QCYCLE=0,
 KRND=6,
 FRMR=0.0,
 DELT=10,
 ISOT=0,
 OUTFRAC=0.9,
 NDUMP=0,
 RANDOM=1,
 OUTDT=30,
 VEGHT=0.5,
 NTURB=0,
 ICONVECT=0,
 ZICONTROLTF=0,
 WINDERRTF=0,
 IVMAX=8,
 VARSIWANT='time','indx','lati','long','zagl','zsfc','foot','mlht',
 /
```

R scripts to Run STILT

It is a lot of work to manually modify the `CONTROL` and `SETUP.CFG` files for each run! Hence a layer of code written in a higher-level language has been designed and constructed to interact with the core Fortran model. These scripts are written in the R language (OpenSource implementation of the S-statistical analysis language). The source code, installation files, and documentation on R can be downloaded from the R Project for Statistical Computing: http://www.r-project.org/ . R is a fully-functional, free, and open-source software that runs on multiple platforms (e.g., Mac OS, Windows, Linux). Furthermore, R is gaining in popularity due to the powerful data analysis packages that have been contributed by users all over the world.

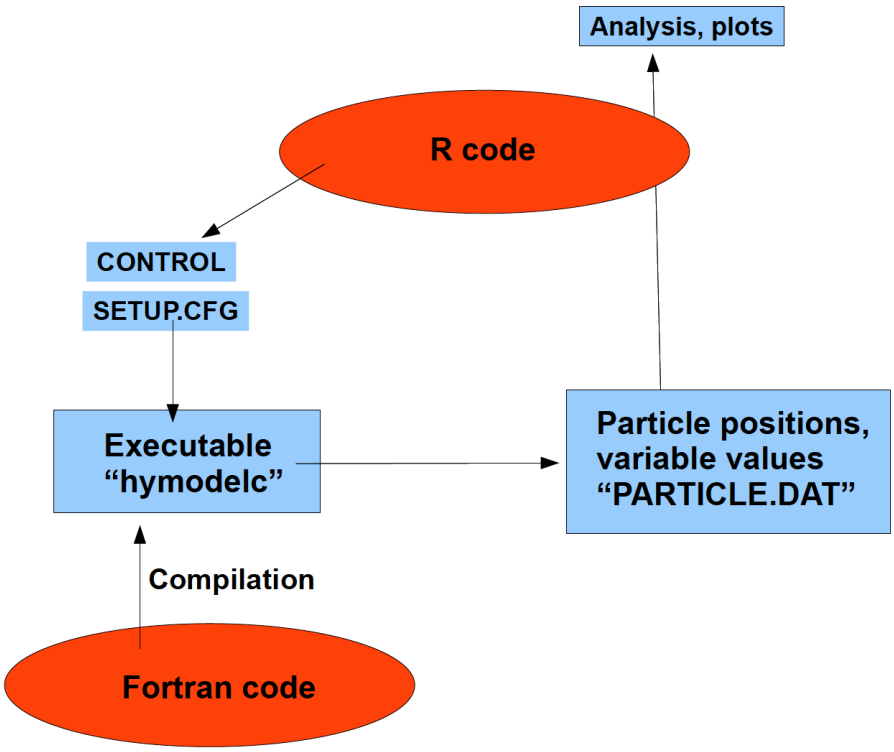

Figure 1: A schematic of how the Fortran code and the R code interact with one another in STILT.

In the Rsc/ directory are scripts to run STILT and to calculate tracer concentrations at the particle location ("receptor") using a "receptor-oriented framework". A description of the receptor-oriented framework employing STILT is:

Gerbig, C., J.C. Lin, S.C. Wofsy, B.C. Daube, A.E. Andrews, B.B. Stephens, P.S. Bakwin, and C.A. Grainger, Toward constraining regional-scale fluxes of CO_2 with atmospheric observations over a continent: 2. Analysis of COBRA data using a receptor-oriented framework, *Journal of Geophysical Research*, *108* (D24), 4757, doi:10.1029/2003JD003770, 2003.

See the file '`Rsc/00README.TXT`' for a description of how to run the R scripts. In particular, the function '`Trajec`' can be used in R to automatically generate the `CONTROL` and `SETUP.CFG` configuration files, call the Fortran executable `hymodelc`, and read in the output (`PARTICLE.DAT`).

In addition, there's a short tutorial '`Rsc/0stilt_tutorial.r`' that generates sample output and illustrates how STILT can be run with the R scripts.

Here a few of the core functions are described:

I. Trajec

`Trajec` is the workhorse function that runs STILT and then reads in the output file `PARTICLE.DAT` By default, the function stores the results in a "`.RData`" file with a name that reflects the starting time and location of the run. For instance, "`.RData2002x08x12x18x42.54Nx072.17Wx00030`" stores the output from a run that started on Aug. 12th, 2002, at 1800 UT. The starting locations is at 42.54-deg N and 72.17-deg W, at a height of 30-m AGL.

```
info<-Trajec(yr=2,mon=8,day=12,hr=18,lat=42.53572,lon=-72.172,agl=30,nhrs=-2,
     numpar=5,delt=10,outdt=30,metlib="/bluejay/stilt/Metdata/",
     metfile=c("edas.subgrd.julaug02","fnl.nh.julaug02"),
     varsout=c("time","index","lat","lon","agl","grdht","foot","zi"),
     rundir="/home/jcl/STILT/Exe/")
```

`.RData` files are in binary format, but are compatible across platforms!

The output from `Trajec` is written to an external file .RData
"info", then, stores the details of this run.

```
> info
                                     yr                                     mon
                                   "2"                                     "8"
                                    day                                      hr
                                   "12"                                    "18"
                                   lat1                                    lon1
                              "42.53572"                                "-72.172"
                                   agl1                                    nhrs
                                   "30"                                    "-2"
                                   delt                                 numpar
                                   "10"                                     "5"
                                  ndump                                 random
                                    "0"                                  "TRUE"
                                 outdt                                   veght
                                   "30"                                   "0.5"
                                 metlib                                   metd1
                "/bluejay/stilt/Metdata/"                                  "edas"
                           doublemetfiles                                metfile1
                                  "FALSE"                  "edas.subgrd.julaug02"
                                metfile2                                   nturb
                        "fnl.nh.julaug02"                                 "FALSE"
                                 outfrac                                    conv
                                   "0.9"                                  "FALSE"
                                varsout1                                varsout2
                                  "time"                                  "index"
                                varsout3                                varsout4
                                     "lat"                                   "lon"
                                varsout5                                varsout6
                                     "agl"                                  "grdht"
```

```
                              varsout7                                varsout8
                                "foot"                                    "zi"
                              nummodel                                 outname
                                   "0"  "2002x08x12x18x42.54Nx072.17Wx00030"
                               outpath                               overwrite
                                    ""                                  "TRUE"
xedas.subgrd.julaug02xfnl.nh.julaug02                                  status
                                   "4"                                     "1"
```

II. getr and assignr

To retrieve the `.RData` files, the getr function can be used. Here is a sample command retrieving the output from a `Trajec` run:

```
dat<-getr(info["outname"],path="./")
```

The function "gets" the contents of `info["outname"]`—i.e., `2002x08x12x18x42.54Nx072.17Wx00030`—from its `.RData` file (`.RData2002x08x12x18x42.54Nx072.17Wx00030`), which can be found in the directory given by "path".

Conversely, assignr creates a `.RData` file with a specified name. Here is an example:

```
assignr("stiltrun",value=dat,path="../",printTF=T)
../.RDatastiltrun created
```

III. Trajecfoot

`Trajecfoot` generates the "footprints" of the receptor, based on the `Trajec` run. Footprints provide the sensitivity of a concentration measurement to upwind fluxes. The unit of the footprint calculated by `Trajecfoot` is in [ppmv/umole/m2/s]. In other words: the footprint gives the concentration change at the starting point of the STILT run (receptor), given an unit flux from the ground surface. This is based on how many particles are found near the ground at a location and how long they reside there.

The following is some sample code to generate the footprint of a receptor:

```
#1) Call Trajec to run
info<-Trajec(yr=2,mon=8,day=12,hr=18,
             lat=43.4516,lon=-80.49242,agl=30,nhrs=-48,
             numpar=100,delt=10,outdt=30,metlib="/bluejay/stilt/Metdata/",
             metfile=c("edas.subgrd.julaug02","fnl.nh.julaug02"),
             ,varsout=c("time","index","lat","lon","agl","grdht","foot","zi"),ru
             ndir="/home/jcl/STILT/Exe/")

#2) Set up input variables to Trajecfoot
foottimes<-c(0,24,48)           #vector of times (backtimes) in hours between
which footprint is computed
zbot<-0                         #lower vertical bound for influence projection,
in meters agl
ztop<-0                         #upper vertical bound for influence projection,
in meters agl
                                #   if ztop set to zero, *surface* influence
will be calculated
lon.res<-1/10                   #resolution in degrees longitude
lat.res<-1/10                   #resolution in degrees latitude
```

```
numpix.x<-40/lon.res            #number of pixels in x directions in grid
numpix.y<-25/lat.res            #number of pixels in y directions in grid
lon.ll<--100                    #lower left corner of grid
lat.ll<-30                      #lower left corner of grid
ident<-info["outname"]   #ident is identifier flag for an
"2002x08x12x18x42.54Nx072.17Wx00030"

#3) Call Trajecfoot
foot<-Trajecfoot(ident=ident,pathname="./",foottimes=foottimes,
                 zlim=c(zbot,ztop),fluxweighting=NULL,coarse=1,
                 numpix.x=numpix.x,numpix.y=numpix.y,
                 lon.ll=lon.ll,lat.ll=lat.ll,lon.res=lon.res,lat.res=lat.res)

#4) Process output from Trajecfoot
xfoot<-apply(foot,c(1,2),sum)   #(a) generate *time-integrated* footprint
xlon<-as.numeric(dimnames(xfoot)[[2]]);xlat<-as.numeric(dimnames(xfoot)[[1]])
xfoot.log<-log10(xfoot);xfoot.log[is.inf(xfoot.log)]<-NA

#5) Generate map of footprint
library(maps);library(fields)
X11();image.plot(x=xlon,y=xlat,z=t(xfoot.log),cex.lab=1.5,cex.axis=1.5,xlab="Lon
gitude",ylab="Latitude")
map("state",add=T);map("world",c("Canada","Mexico"),add=T)
title(main=ident,cex.main=1.5)
```

The resulting map is shown here:

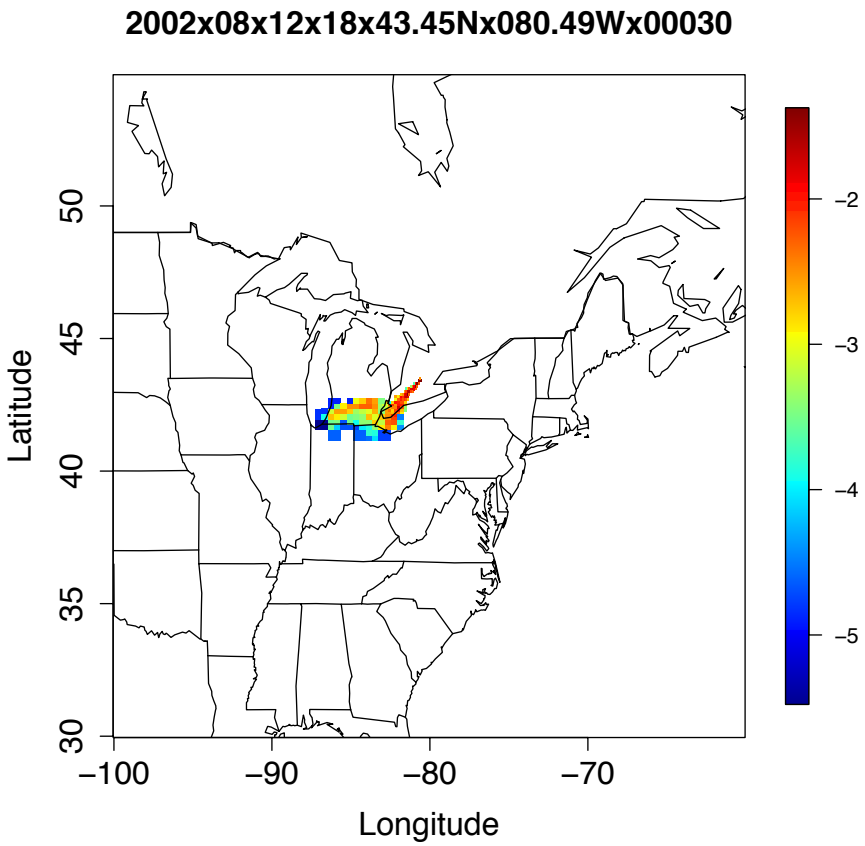

Figure 2: Footprint map produced by Trajecfoot. The color scale is logarithmic (base 10).