Appendix A
Running the OZIPR Model

Introduction

The OZIPR model is an air-quality simulation model that runs on a personal computer. It is a single-day trajectory model that was intended to focus on the atmospheric chemistry that leads to ozone formation. Specific details of the chemistry and of the scenario to be modeled are read as inputs, and so can be varied by the user. For the current application the chemistry is a version of the SAPRC97 chemical mechanism, modified to focus on the chemistry relevant to the production of hazardous air pollutants (HAPs), specifically formaldehyde, acetaldehyde, and acrolein.

To run OZIPR, the user needs the following:

1. The executable program code named OZIPR.EXE
2. A file detailing the chemical mechanism (CAL97.MEC for the current application)
3. A zenith angle file (ZEN.CAL for the current application)
4. A specific input file detailing each scenario to be run

The OZIPR model was designed to be quite flexible, with the user controlling the specific details of the model run through an *input file.* All of the model runs done for this project (e.g., all study areas, seasons, counties, day types) used the same compiled program code; the only difference between runs was in the input file.

The OZIPR model is run from the command line (or from a batch file) with one argument, the name of the input file specifying the data for that run. OZIPR input files are distinguished by the file name extension .INP. For example, the command

```
ozipr.exe houston.inp
```

will run the model using a file named HOUSTON.INP as the input file. These commands and arguments are not case sensitive. At the end of the run, a standard output file is created with the same root file name as the input file, but with extension .OUT (HOUSTON.OUT for the example above).

Optionally, if the user wishes to run many scenarios, a batch file can be set up to carry out multiple model runs and to do file manipulation and post-processing. This method of running OZIPR was used in the current application, but is not discussed here in detail. The information in the batch file is unique to each computer (depending on directory structure and the file names used). The results are the same as for the model runs carried out individually.

A more general explanation on how to run the model is provided by the *User's Guide for Executing OZIPR* (Gery and Crouse, 1990). The information provided below explains the structure of the input file used for the current application—the estimation of primary and secondary HAP production. Note that even for this specific task, there are many ways of structuring the input file.

The OZIPR Input File

The OZIPR input file is a text file consisting of a series of *options* and *commands*. An example input file is provided that was used to model Houston. This input file is examined line by line in this section, so that users can understand how to prepare similar input files for their own scenarios. Actually, the current project required a total of 48 input files for Houston (and for each of the other study areas), to cover various combinations of seasons, day types, urban versus rural, and primary versus secondary HAP formation.

The task of physically assembling and editing a large number of input files can be quite onerous. The OZIPR model has a feature that allows the input file to contain references to other files (indicated by an @ sign), which means that the contents of the other file are copied verbatim into the input file at the location of the @ sign.

The use of the @ sign has several advantages. First, for input data that is universal but lengthy such as the chemical mechanism (which contains several pages of information covering more than 100 reactions), the indirect file reference saves space. To document model runs, it is necessary to save (and usually to print) the input files; if there are hundreds of model runs then there would be hundreds of copies of the same chemistry, if the @ sign were not used. Furthermore, if for any reason the chemical mechanism needed to be updated, by using the @ sign only one file would have to be altered, whereas the same change would have to be made in every input file if the mechanism were explicitly listed in each one.

The @ character must appear in the first column of a line and is followed immediately by a file name. Standard DOS file-naming conventions are used; a path can be specified if the file is not in the current directory. Also, the file name should not exceed eight characters in length. Do not enclose the name in quotes, and do not write anything else on that line of the input file.

The file named after the @ sign should contain simple ASCII text. The contents are copied into the input file at the point marked by the @ sign. This is similar to the use of *include* statements in Fortran, or *macro variables* in SAS. Once all the @ signs have been replaced by the text to which they refer, the resulting input file must satisfy certain rules of syntax.

General rules for the input file are:

1. Only the first 80 characters (including blanks) per line are read by OZIPR; longer lines are truncated.
2. Only the first four letters of a command option name are significant. The remainder are ignored by the program but may be spelled out to help the user read the input file.
3. Comments can be included following the first character, !, or bracketed between { } or ( ).
4. Top-level commands are followed by a > symbol and end with a < symbol.

5. The commands are not case sensitive.
6. White space and indentation are not important; these can be adjusted for convenience. (But note that an @ command must begin in column 1.)
7. Top-level assignments (containing an = sign) end with a semicolon.
8. Certain assignments include a list of data or several lower level assignments; these are separated using commas.
9. The input file must terminate with END.

There are some restrictions on the order of commands. These are indicated in more detail in the OZIPR user's guide. For users who wish to submit runs similar to the sample application, it is best to keep the commands in the same order to avoid problems. Otherwise, the user can reformat the sample file by adding, modifying, or deleting options as necessary. Optional commands that are not used in the sample file are not discussed here.

SAMPLE OZIPR INPUT FILE FOR HOUSTON

```
@..\cal97.mec
@..\zen.cal
TITLE > Houston - Spring - Day #1 - Urban - Carbonyls <
PLACE >
   CITY = Houston;
   LAT  = 29.8;    LON   = 95.2;  TZONE = 6;
   YEAR = 1996;    MONTH = 4;     DAY   = 15;
< (PLACE)
TIME > 0800, 2000 <
BOUNDARY >
   REAC =
      ALK4,    0.2794,   0.2794,   0.2794,
      ALK7,    0.2073,   0.2073,   0.2073,
      ETHE,    0.0356,   0.0356,   0.0356,
      PRPE,    0.0191,   0.0191,   0.0191,
      TBUT,    0.0721,   0.0721,   0.0721,
      TOLU,    0.0511,   0.0511,   0.0511,
      XYLE,    0.0333,   0.0333,   0.0333,
      TMBZ,    0.1018,   0.1018,   0.1018,
      RCHO,    0.0090,   0.0090,   0.0090,
      BDIE,    0.0000,   0.0025,   0.0025,
      HCHO,    0.0000,   0.0200,   0.0200,
      ALD2,    0.0000,   0.0200,   0.0200,
      ACRO,    0.0000,   0.0010,   0.0010,
      NRHC,    0.1913,   0.1478,   0.1478;
   IFRACTION NO2 = 0.20;
   TRANSPORT=
@..\aloft\housprA.txt
   INIT =
      HONO = 0.001;
   DEPO [12] =
      NO2  = 0.48,0.54,0.60,0.60,0.60,0.60,0.60,0.60,0.54,0.48,0.36,0.24,
      O3   = 0.60,0.70,0.80,0.80,0.80,0.80,0.80,0.80,0.70,0.60,0.50,0.30,
      HNO3 = 3.00,3.30,3.50,3.50,3.50,3.50,3.50,3.50,3.30,3.20,3.00,2.60,
      H2O2 = 1.80,1.90,2.00,2.00,2.00,2.00,2.00,2.00,1.90,1.80,1.70,1.70,
      PAN  = 0.48,0.54,0.60,0.60,0.60,0.60,0.60,0.60,0.54,0.48,0.36,0.24;
```

```
< (BOUNDARY)
@..\met\hou_spr1.txt
@..\emis\houspruc.txt
CALCULATE >
@..\init\houspru.txt
   PRINT[CONC] =
      NAMES[5] = CO,O3,HCHO,ALD2,ACRO,
      AVG[5]   = CO,O3,HCHO,ALD2,ACRO;
< (CALCULATE)
END.
```

Structure of the Sample Input File

The sample input file follows certain stylistic conventions that are not essential but add to clarity. In particular, commands like TITLE whose content can fit entirely on one line are written as:

```
COMMAND > content <
```

Multiline commands are written as:

```
COMMAND >
        content line 1
        content line 2
        ...
        content line n
< (COMMAND)
```

Both of these cases follow the rule that the command begin with > and end with <. In the multiline case, a comment is added after the < symbol merely to clarify to which command it applies.

Within a first-level command such as PLACE or BOUNDARY, the lower level commands are indented three spaces. This is arbitrary but also adds clarity. Subcommands are indented a second time in a similar manner. The user should ensure that indentation does not push the content of the line beyond column 80, or it will be truncated when it is read into OZIPR.

The sample file discussed below is one of the files used to model HAP production in Houston. References below to “this project” indicate options and data values that were used in this application. Many of the OZIPR commands allow additional data to be specified. These unused options are not addressed further in this section; if interested in using them, the user should refer to the OZIPR user’s guide.

A line-by-line description of the sample input file follows. The printout of the sample file has line numbers added for clarity. These numbers should not appear when using the file; the first @ sign on line 1 must be in column 1, with the other lines in the same relative position.

Line 1: **@..\cal97.mec**

The @ sign indicates that the contents of the file named CAL97.MEC are to be inserted here. The “..\” tells the program that the file is located in the parent directory (i.e., one level higher than the current level). The CAL97.MEC file contains a modified version of the SAPRC 97 mechanism. The modifications are primarily concerned with reactions that create or destroy acrolein, which is one of the HAPs of interest in this simulation. Details of the chemical mechanism are given elsewhere. The same chemical mechanism is used for all study areas and all seasons under this project, so this file does not need to be modified for new scenarios.

Line 2: **@..\zen.cal**

This file contains information on the zenith angle dependence of the chemical reaction rates. There are 18 different functional dependencies specified. The file ZEN.CAL is just the version used with the original SAPRC97 mechanism, without modifications. It was used for all model runs on this project.

Line 3: **TITLE > Houston - Spring - Day #1 - Urban - Carbonyls <**

This is the title to be printed with the output data. It can be anything, up to 72 characters. The > and < signs are used to indicate the beginning and end of the TITLE command.

Lines 4–8: **PLACE >**

These lines define the location of the study area. PLACE is used in determining the zenith angle of the sun throughout the day. CITY name can be up to 24 characters. LAT is latitude in decimal degrees north, and LON is the longitude in decimal degrees west. TZONE is the local time zone in hours behind Greenwich mean time. Unlike the description in the OZIPR user's manual, the sample run uses standard rather than daylight saving time. For Houston, which is in the central time zone, TZONE = 6 (other values are Eastern=5, Mountain=7, Pacific=8). YEAR is the year modeled, MONTH is the number for month, and DAY is the day of the month. In this example the solar elevation is calculated for April 15.

Line 9: **TIME > 0800, 2000 <**

This specifies the start and stop time for the run in military time. In this case, the start time is 8 a.m. and the stop time is 8 p.m. (20 hours, 0 minutes). Local standard time is used for these runs. The same start and stop times were used for all runs under this project.

Line 10: **BOUNDARY >**

This command is the beginning of a block of data specifying various aspects of the boundary conditions (chemical concentrations) for the model run. The BOUNDARY command contains the following subcommands: REAC, IFRACTION, TRANSPORT, INIT, DEPO. These are discussed in turn.

Lines 11–25: **REAC=**

The REAC data consists of a table with four columns. There is one row for each chemical species (14 in this case). The list of species must match the list given under the CNUM option in the MECH command (this is part of the CAL97.MEC file). The first column gives the species name. The last three columns indicate the partitioning of the total VOC into components (fractions), which must total to unity. The first data column represents the VOC in the hourly emissions, the second the composition of the aloft VOC, and the third the transported surface-layer VOCs. In the current application there are no transported surface-layer VOCs, so the third column is not used (but must be present anyway). Since specific hourly emissions of the target HAPs are provided as input under the MASS option, these HAPs should not be present in the first column of the REAC data or else the emissions would be double counted. The meaning of the species names are as follows:

ALK4 – Alkanes with 3, 4, and 5 carbons (primarily butane and isopentane)

ALK7 – Heavier alkanes (6 or more carbons)

ETHE – Ethene (also known as ethylene)

PRPE – Propene (also known as propylene)

TBUT – Heavier alkenes (4 or more carbons)—primarily trans-2-butene

TOLU – Toluene (also ethylbenzene)

XYLE – Xylene

TMBZ – Trimethylbenzene (and heavier aromatics)

HCHO – Formaldehyde

ALD2 – Acetaldehyde

RCHO – Heavier aldehydes

ACRO – Acrolein

BDIE – 1,3-Butadiene

NRHC – Nonreactive hydrocarbons

For the current project the REAC data were city-specific but not seasonally dependent.

Line 26: **IFRACTION = 0.20;**

This sets the initial ratio of NO_2/NO_X at the start of the run. All of the runs under this project used the same setting.

Lines 27–28: **TRANSPORT =**

The TRANSPORT command sets the concentrations of certain species in the air aloft (above the mixing layer), and optionally also in the initial surface mixture, if this is not due to local emissions. In the sample file, the data is contained in the file HOUSPRA.TXT, which contains the lines

```
VOCALOFT = 0.1000,
NOXALOFT = 0.0003,
O3ALOFT  = 0.0300,
COALOFT  = 0.2000;
```

These statements set the concentrations of VOC, NO_X (i.e., NO + NO_2), ozone, and carbon monoxide in the aloft air. The VOC is in units of ppmC, the others are in ppm. This air will be mixed into the surface layer as the day progresses and the mixing height increases. The data were written on a separate file to aid in the automated generation of multiple input files, but the user could simply write the four lines above in place of the file reference.

Lines 29–30: **INIT =**

Here the initial concentration of chemicals not listed elsewhere can be specified. In this case, nitrous oxide (HONO) concentration is specified in ppmV. Since the target HAPs are not considered to be part of the VOC partitioning, the 8 a.m. concentrations must be entered here as well. The following is an example:

```
INIT =
   HCHO = 0.00600,
   ALD2 = 0.00600,
   ACRO = 0.00030,
   BDIE = 0.00075,
   HONO = 0.001;
```

The first three species are the target HAPs: formaldehyde, acetaldehyde, and acrolein. Generally, these initial carbonyl values should be around 50% of the levels present in the evening (meaning at the end of the run). This 50% value is typical of the ratio seen in field data. In most scenarios the hourly emissions and the photochemistry dominate after the first couple of hours, so the concentrations seen through most of the day are not strongly dependent on the initial conditions.

Lines 31–36: **DEPO=**

This subsection contains deposition velocities for selected species. Each species is assigned a vector of hourly values (in cm/s), one value for each hour covered in the simulation.
These values were kept the same for all model runs under this project.

Line 37: **< (BOUNDARY)**

This indicates the end of the BOUNDARY command.

Line 38: **@..\met\hou_spr1.txt**

This file contains the following text:

```
MET >   {HOUSTON,SPRING 1 }
   DILUTION=
      MHINIT  =   617,
      MHFINAL =  1081;
   TEMPERATURE[13,C] =
      21.1, 22.2, 23.3, 24.4, 25.0, 25.6,
      26.1, 26.1, 25.6, 25.0, 23.9, 22.8, 22.2;
   PRESSURE[AT] = 1.00;
   RH[13] =
      90.0, 84.0, 77.0, 72.0, 67.0, 66.0,
      65.0, 64.0, 67.0, 70.0, 74.0, 79.0, 84.0;
< (MET)
```

The meteorology section contains four subsections. The first is DILUTION, which is used to determine mixing height during the run. This subsection contains the morning and afternoon mixing heights (in meters). The second subsection is TEMPERATURE, which contains hourly temperature values for the model run. The keyword is followed by two arguments in square brackets, i.e., TEMPERATURE [nval,units]. The first is the number of values, which should be one more than the number of hours in the simulation. This is because the values represent point measurements at the beginning and end of each hour, not hourly averages. The second argument is the units; use “F” for Fahrenheit, “C” for Celsius, or “K” for Kelvin, which is the default. If degrees Kelvin are used, then the square brackets only need to contain the number of values. After the square brackets comes an equal sign, followed by the hourly values separated by commas.

The third subsection is PRESSURE. The square brackets contain the units. The default units are “AT,” for “atmospheres.” Only one value is given for pressure to be used throughout the run. This will generally be 1.0 atmosphere except at high elevations, such as Denver.

The fourth subsection is RH, or relative humidity. Like temperature, the square brackets contain the number of values, which should be (# hours + 1), since these are also point measurements at the beginning and end of each hour. The values are in percent (so 50 is 50% relative humidity).

Note that the OZIPR user’s guide is incorrect in stating that the keyword *METEOROLOGY* can be used for this section. Only the three-letter word *MET* will work.

Line 39: **@..\emis\houspruc.txt**

This section contains the hourly emission rates for various species. The file HOUSPRUC.TXT, which stands for “HOUston SPRing Urban with Carbonyls,” contains the following text:

```
MASS [12] >
  VOC = 0.3000,
      8.52, 8.89, 9.39, 9.93,10.35,10.57,10.95,11.41,11.54,11.27, 8.38, 6.10,
  NOX = 0.0300,
      3.99, 3.92, 4.04, 4.19, 4.27, 4.24, 4.32, 4.56, 4.65, 4.60, 3.95, 2.14,
  CO  = 0.8000,
      26.8, 29.8, 31.0, 32.5, 33.4, 33.1, 34.0, 36.5, 37.7, 37.4, 30.9, 15.8,
  HCHO[30] = 0.0000001,
     .0396,.0403,.0412,.0423,.0430,.0427,.0434,.0452,.0461,.0458,.0395,.0216,
  ALD2[44] = 0.0000001,
     .1397,.1481,.1543,.1621,.1667,.1651,.1701,.1831,.1900,.1879,.1577,.0931,
  ACRO[56] = 0.0000001,
     .0074,.0074,.0074,.0074,.0074,.0074,.0074,.0074,.0074,.0074,.0074,.0074,
  BDIE[54] = 0.0000001,
     .0259,.0259,.0260,.0259,.0259,.0259,.0259,.0259,.0259,.0259,.0259,.0259;
< (MASS)
```

The number in square brackets after the word *MASS* indicates the number of hourly values. For each chemical species three types of data are presented. On the left of each equal sign the chemical species is named, followed by square brackets enclosing the molecular weight of the species. Molecular weights are not needed for the first three species (VOC, NO_X, and CO), since defaults are used. The first value after the equal sign represents the initial concentration of the species at the start of the run. (For VOC, NO_X, and CO, these also appear in the CALC option described below.) The remaining values represent hourly emissions in units of kg km^{-2} h^{-1}.

The first value (concentration) is distinguished from hourly emissions by putting it on a separate line. The hourly emissions represent the sum of anthropogenic plus biogenic emissions and vary by season. The number of these hourly values must equal the number specified after the keyword *MASS*, and must be the same for all species (in this case, 12 values are supplied per species). The values are separated by commas except the last one for the last species, which is followed by a semicolon to indicate that no more species will be listed. For VOC this mass represents the total VOC mass. It is apportioned into its constituent species according to the fractions specified in the REAC section of the BOUNDARY command. For model runs without primary carbonyl emissions, the last three species (HCHO, ALD2, and ACRO) are removed from the list, and the last hourly value for BDIE is followed by a semicolon.

Line 40: **CALCULATE >**

The CALCULATE command contains two parts. The first part,

Line 41: **@..\init\houspru.txt**

reads data from the file HOUSPRU.TXT, and assigns the initial concentrations for VOC, NO_X, and CO. The VOC concentration is in units of ppmC, the other two are in ppmV. The file HOUSPRU.TXT contains the following text:

```
VOC = 0.300;
NOX = 0.030;
CO  = 0.800;
```

These values should agree with the values specified under the MASS section described above.

Lines 42–44: **PRINT[CONC]** =

The PRINT command controls the details of the OZIPR output. The expression [CONC] after the word *PRINT* requests the printing of hourly concentrations. Two other options are specified under PRINT. The first option (NAMES) gives the list of chemical species to be printed in the output file. The second option (AVG) specifies the chemical species for which summary statistics are printed. For both NAMES and AVG, the number of species is specified in square brackets to the left of the equal sign, followed by a list of species to the right. In addition to this list, OZIPR also prints values for the total VOC and for NO_x.

Line 45: **< (CALCULATE)**

This signals the end of the CALCULATE command.

Line 46: **END.**

The final line in the OZIPR input file is the mandatory **END.** command, which signifies that there are no more input commands to be processed.

Preparation of Multiple OZIPR Input Files

The present study required the preparation of up to 480 input files (10 study areas times 48 scenarios each—four seasons, three days each, urban/rural, primary/no primary carbonyls). If all these files were prepared individually by hand then it would not only be time consuming, but the chance of making an error would be fairly high. To get around this, a SAS program was written to generate OZIPR input files automatically. Study-specific data such as the city name, location, and time zone are entered into the SAS program, and a series of 48 OZIPR input files are generated. These 48 files do not directly contain scenario-specific emissions and meteorology; instead they refer to external files that contain the data. (OZIPR input files can refer to external files by use of the @ sign.) Two other SAS programs were written, one to prepare all the necessary emissions files (for all 10 study areas at once), and the other to prepare all necessary meteorological files. Apart from the REAC files described in Section 3, these three SAS programs set up all the input data and files needed for all the OZIPR model runs.

The OZIPR input files are named according to the following system. (File names must follow the DOS convention and cannot exceed 8 characters, plus a 3-character extension.) The first three characters identify the study area (ATL=Atlanta, BOS=Boston, CHI=Chicago, DEN=Denver, HOU=Houston, LAX=Los Angeles, PHX=Phoenix, PIT=Pittsburgh, SEA=Seattle, WDC=Washington, DC). The fourth and fifth letters identify the season (WI=winter, SP=spring, SU=summer, AU=autumn). The sixth character in the file name is a number representing the meteorological cluster (1, 2, or 3). The seventh character is "U" for urban areas or "R" for rural areas. The eighth character in the file name is "C" if the run includes primary carbonyl emissions and "N" if it does not. Thus, the example file named HOUSP1UC.TXT is a spring run for cluster

1 for the urban Houston area, with carbonyl emissions. Note that file names are not case sensitive and may appear as lowercase.

The OZIPR program runs very quickly (about 2 seconds per scenario) on a modern personal computer. Batch files were created to carry out all 48 runs for a city and to concatenate the resulting output tables. The batch files are not necessary but simply increase the efficiency when a large number of model runs are to be conducted.