Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/6003--14-9493

Acute Exposure Guideline Levels (AEGLs) for Time Varying Toxic Plumes

J. P. Boris
G. Patnaik

Laboratories for Computational Physics and Fluid Dynamics

September 12, 2014

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
12-09-2014	Memorandum Report	

4. TITLE AND SUBTITLE

Acute Exposure Guideline Levels (AEGLs)
for Time Varying Toxic Plumes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

J.P. Boris and G. Patnaik

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
64-4464-03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6003--14-9493

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)

NRL

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Toxic airborne contaminants, when inhaled, can cause adverse health effects. The U.S. EPA tabulates three Acute Exposure Guideline Levels (AEGLs) for predicting the onset of these adverse effects from specific length exposures to specific concentrations of toxic chemicals for a general population. Inhalation exposures in the real world, however, vary strongly in space and time and thus do not correspond to the specific durations and exposure levels tabulated by the EPA–a complication that EPA does not resolve. This report gives an accurate, science-based algorithm with fast, easy-to-use software to fill this gap. The software package, called EAGLE, predicts the onset of AEGL 1, 2 and 3 conditions for an arbitrary airborne agent time history supplied by the user. EAGLE can be used with field data, sensor data, wind tunnel measurements, Computational Fluid Dynamics models and rapid-response plume prediction software to convert agent time histories into meaningful actionable health effect predictions. It can also be used to predict the strong effects of realistic concentration fluctuations and to compare plume prediction models in a meaningful way.

15. SUBJECT TERMS

Toxic airborne contaminants
Health effects prediction
Space and time varying exposures
Extension of EPA AEGLs

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			J. P. Boris
Unclassified	Unclassified	Unclassified	Unclassified Unlimited	29	**19b. TELEPHONE NUMBER** *(include area code)* (202) 767-3055

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Acute Exposure Guideline Levels (AEGLs) for Time Varying Toxic Plumes

I. Introduction

The U.S. EPA has established three Acute Exposure Guideline Levels (AEGLs) for predicting the onset of adverse health effects from inhaling specific duration exposures to specific concentrations of toxic chemicals for a general population,. Most of this material is openly available on the internet (e.g. http://www.epa.gov/oppt/aegl, 2013). Further, a number of organizations, agencies and nations use these guidelines and may even mandate this approach to standardize estimating and reporting toxic health effects. The main drawback of the AEGLs is that only a few fixed-duration exposures to a few constant-density conditions are tabulated. The issue of how to treat real toxic plumes, whose agent density varies strongly and often quickly in space and time, is left undefined. Atmospheric transport and dispersion models provide estimations of a plume's movement and the associated agent density variations in 2D and 3D, but most crisis managers would like to see these results expressed in terms of the likely health effects – and get the information while there is still time to respond to and hopefully mitigate the threat.

We have developed a new algorithm and software to compute the onset of AEGL 1 (notable discomfort), AEGL 2 (irreversible or serious adverse impact), and AEGL 3 (life threatening or death) health-effects in response to general, time-varying concentration profiles that usually don't satisfy the specific constant-density-over-extended-period conditions expressed in the tabulated AEGLs. The AEGL 1, 2, or 3 onset condition for any agent concentration history can be computed without reference to any particular pre-tabulated concentration or exposure duration. Plumes over entire urban areas can be integrated in fractions of a second, making this software implementation well suited to use with a dispersion model or CFD code.

The 'new' method reported here is based on an application of the 'Induction Parameter Model' developed in 1980 for fast combustion and explained in detail in the book "Numerical Simulation of Reactive Flow", (Oran and Boris, 2001, 2nd edition). The present application to potentially toxic plumes uses the ten Berge generalization of Haber's law (e.g. Stage, 2004; ten Berge, *et al.,* 1986) integrated against a given contaminant concentration time history. The new calculation is very simple, very fast, and can be applied to any chemical or agent for which the AEGL onset conditions are tabulated.

We describe this software package, which we are calling EAGLE, in the sections below. The package contains the tabulated AEGL data for chlorine (CL_2) and ammonia (NH_3) at 20^0C and one atmosphere and provides a third generic, power-law contaminant (PLC) that allows the user to choose a simple toxic loading rate power law and investigate how fluctuations in the agent density couple to nonlinearity in the toxicity to determine health effects. PL = 1.0, for example, corresponds to an integrated dose-dependent toxicity.

After a short discussion of the Acute Exposure Guideline Levels (Section II), this report describes the induction parameter algorithm implemented in EAGLE (Section III) and the software being provided (Section IV). In Section V, we present several tests that a user can repeat to help understand how this Fortran package works and how it can be used in analyses and in conjunction with dispersion models. Section VI considers some consequences of this new AEGL health-effects onset capability applied to realistic time-varying contaminant plumes in an urban setting. It is found, not unexpectedly, that the non-linear behavior of toxic-loading

Manuscript approved August 26, 2013.

formulae considerably increases the AEGL 1, 2 and 3 exposure hazard areas when contaminant agent concentrations fluctuate naturally (e.g., Bogen and Gouveia, 2008).

We include a Fortran version of the programs as Appendices A and B. The software is designed so that a user can enter tabulated AEGL data for any agent or chemical for which the data are available or can be approximated. Another consequence is that AEGL 1, 2, and 3 onset times and hazard areas can be determined without overly conservative reference to one of the specific AEGL exposure durations (10 min, 30 min, 60 min, 4 hours, or 8 hours).

II. The EPA Acute Exposure Guidelines (AEGLs)

The EPA Acute Exposure Guideline Levels (AEGLs) are one official method for gauging health effects from exposure to toxic chemicals. Figure 1 below is the EPA AEGL table for chlorine (www.epa.gov/oppt/aegl/pubs/results56.htm). These guidelines are tabulated for specific time intervals, 10 minutes, 30 minutes, 60 minutes, 4 hours, and 8 hours. An inhaled exposure to the specified concentration, held constant for the specified duration, is expected to cause onset of the AEGL 1 (notable discomfort), AEGL 2 (irreversible or serious adverse impact), and AEGL 3 (life threatening or death) health effects in a general population. Parts per million (ppm) is an expression of a concentration; the corresponding density depends on temperature. In EAGLE these concentrations are converted to mg/m^3 densities, the temperature-independent units usually computed directly by CFD models. The formula for this conversion in EAGLE is

$$mg/m^3 = \frac{MolecularWeight * ppm}{24.04}$$ for one atmosphere at 20^0C.

The molecular weight for chlorine (CL_2) is MW = 70.9 and for ammonia (NH_3), MW = 17.03.

Figure 1 just below reproduces the chlorine table from the web. Note that AEGL 1 has a single threshold concentration 0.5 ppm for all of the tabulated exposure durations. The information does not indicate how quickly the symptoms set in but certainly this happens faster than 10 minutes. The effective power law is PL = 0.0 (or its inverse – infinity).

Chlorine Results

AEGL Program

Chlorine 7782-50-5 (Final)					
ppm					
	10 min	30 min	60 min	4 hr	8 hr
AEGL 1	0.50	0.50	0.50	0.50	0.50
AEGL 2	2.8	2.8	2.0	1.0	0.71
AEGL 3	50	28	20	10	7.1

Table 1. AEGL concentration values, in parts per million, for chlorine from the U.S. EPA website. AEGL 1 threshold concentrations are the same for each time, indicating a single threshold for the onset of symptoms independent of the duration. This is a limiting behavior for nonlinear toxic loading. AEGL 3 concentration values follow a quadratic power law. Doubling the concentration reduces AEGL 3 onset time by a factor of four. AEGL 2 values behave as a quadratic for low concentrations and switch to a threshold density for concentrations above 2.8 ppm.

Now consider AEGL 3 as tabulated in Table 1. Doubling the ppm value from 10 ppm to 20 ppm reduces the delay to symptom onset by a factor of four from 4 hours down to 1 hour. The dependence is quadratic and the power-law coefficient is PL = 2.0 to the accuracy with which the numbers are expressed across the whole range of times. Again no information is given about how far this power-law behavior extends to times shorter than 10 minutes or longer than 8 hours. AEGL 2, between AEGL 1 and AEGL 3 in severity, is a combination of these two behaviors with intermediate concentration values. For low concentrations, PL = 2.0 but for concentrations above 2.8 ppm the behavior becomes a threshold, again with no short-time limit expresssed.

Table 2 below shows the same information as Table 1, but for ammonia. The toxicity levels are 50 or 60 times lower than for chlorine, but the behavior with time and density is very similar.

Ammonia Results

AEGL Program

Ammonia 7664-41-7 (Final)					
ppm					
	10 min	30 min	60 min	4 hr	8 hr
AEGL 1	30	30	30	30	30
AEGL 2	220	220	160	110	110
AEGL 3	2,700	1,600	1,100	550	390

Table 2. AEGL concentration values, in parts per million, for ammonia from the U.S. EPA website. AEGL 1 threshold concentrations are the same for each time, indicating a single concentration for the onset of symptoms (e.g. eyes watering) independent of the duration. AEGL 3 concentration values follow a quadratic power law. Doubling the concentration reduces the time to AEGL 3 onset by a factor of four. AEGL 2 values for ammonia show a non-power-law behavior with two plateaus.

III. Algorithm to Evaluate the AEGLs in Time-Varying Plumes

This method is based on a differential representation of the nonlinear ten Berge generalization of Haber's law (ten Berge et al., 1986). A quantity we shall call the 'toxic load' (TL) is integrated from zero using the actual evolving contaminant concentration time history. A good discussion of the background of this problem is given by Stage (2004) who introduces a method based on the linear dosage integral to accomplish some of the same ends. When the time-dependent toxic load, defined by the integral

$$TL_k(t) = \int_0^t \left[\frac{dTL_k(t')}{dt'} \right] dt', \qquad \text{(Eq. 1)}$$

increases from zero to one, the corresponding AEGL condition onset has occurred. In Eq. (1), subscript k = 1, 2, 3 selects the particular AEGL condition. For each AEGL, the onset condition is reached when the distinct $TL_k(t)$ function reaches unity.

Our implementation is based on an application of the 'Induction Parameter Model' developed for fast combustion (Boris, et al. 1980) and explained in detail in the book **Numerical**

Simulation of Reactive Flow (Oran and Boris, 2001, 2nd edition). The algorithm defines a power law for the rate of accumulation of toxic load spanning each of the time intervals for AEGL 1, AEGL 2, and AEGL 3. This I, in essence, an interpolation giving the toxic load accumulation rate as a function of the instantaneous density.

For example, consider the 60-min to 4-hour band for the ammonia AEGL 3 immediately above. At 550 ppm, the rate of toxic-load accumulation, *TLrate*, is 1/14400 per second. At this concentration, multiplying *TLrate* by four hours gives unity. The formula defining *TLrate* throughout this band is

$$TLrate(t) \equiv \frac{dTL}{dt} = \frac{1}{14400\,\mathrm{s}}\left[\frac{\rho(t)}{\rho_{4hr}}\right]^2. \qquad \text{Eq. (2)}$$

Here $\rho(t)$ is the instantaneous density at the point in question and ρ_{4hr} is the 4-hour EPA-tabulated density for the particular AEGL being considered (subscript suppressed in Eg.(2)). The exponent is exactly 2 above because 550 ppm is exactly half of the tabulated 1-hour value of 1100 ppm. An appropriately chosen exponent of the density ratio can span the two rates (inverse times) that define each band - provided that increasing the concentration never actually reduces the toxicity. As earlier, there are really three such equations, one for each of the three AEG Levels.

To complete the algorithm, we define the integrated toxic load 'TL', the 'induction parameter' for this application, as the integral

$$TL(t) = \int_0^t TLrate(t')\,dt'. \qquad \text{Eq. (3)}$$

TL(t) will be integrated numerically in the user's dispersion model or a field trial's data analysis package. When *TL(t)* reaches unity the corresponding AEGL onset criterion (k = 1, 2, or 3) has been reached. Clearly three summands will have to be stored at each result location when all three AEGLs are being considered simultaneously. Rather than further discussing the algorithm here, we will defer some of the details to descriptions of the software in the next section. The source code is being provided and should be easy enough to read and use.

This calculation is simple, very fast, and can be applied to any chemical or agent for which the EPA has tabulated the AEGL onset conditions. This new algorithm is designed for integration against realistic (measured or computed) contaminant density time histories as they are encountered. Thus AEGL hazard areas over entire urban areas can be updated in fractions of a second for use with dispersion models or in emergency incidents. The method does not require knowing the time history before computation can begin so it can be evaluated *in situ* as the plume is evolving. This need not be a post process but does require storing one, two, or three additional variables, the evolving Toxic Loads at each point where a result is desired. The result also will not change discontinuously when a new maximum peak concentration is briefly encountered.

Using a power law defined for each of the individual AEGL bands guarantees that 'EAGLE' will accumulate exactly the threshold toxic load for each AEGL time when the density is artificially held constant at the tabulated threshold level. This simple model agrees implicitly in form with the table values themselves, which also generally satisfy a power law to the tabulated one- or two-digit accuracy. There can be a discontinuity in slope when the exponent changes from band to band but the computed toxic loading rate *TLrate* will be continuous. In any case, to EPA does not provide enough information to remove this slope discontinuity convincingly.

Please note that there is an implicit assumption that the order in which the toxic load accumulates does not matter. One minute at 1100 ppm will have the same deleterious effect whether it is encountered at the very beginning of the exposure, in the middle, or near the end. Other methods also make this implicit assumption and there appears to be no data supporting additional complexity in favor of a smoother algorithm. There are also two other choices we have made to complete the package. For times longer than 8 hours, or more correctly for concentrations (densities) below the 8-hour value, the EPA table implies that no toxic load is accumulated. Some additional information would have to be given to extend the tables provided to the right (lower concentrations and longer times). When there is a non-zero exponent in the last band, as for AEGL 3 in the examples above, that exponent can be used to extend the TLrate evaluation. Choosing zero until some additional toxicological information is available is an option. Such information might be a 24-hour AEGL - or a sensible formula without an artificial discontinuity.

The same problem occurs at the left edge of the table for times shorter than 10 minutes. It is quite clear that very high concentrations will be more toxic than the 10-minute values. One can, in principle, get a lethal exposure in a breath or two so the maximum rate of toxic load accumulation must be faster than 1/600 per second. Therefore we define an extrapolation band between τ_{min} and 10 minutes using the same exponent as computed for the 10-min to 30-min band. It is also clear that this extension has to be limited in some manner. In our applications we choose τ_{min} as 200 seconds, with a maximum allowed concentration (density) computed accordingly. Again the program will show the implementation of these assumptions. The interested user is certainly free to play with these assumptions and to change them.

The new calculation is simple, fast, and can be applied to any chemical or agent for which the AEGL onset conditions are specified. It is found, not unexpectedly, that non-linear behavior in the toxic-loading formulae considerably increases the exposure hazard area when the concentration fluctuates naturally.

IV. Description of the Software Provided

Appendix A reproduces a Fortran file containing the three routines a user will need: subroutine AGENT_SETUP, real function TL_RATE, and subroutine POWER_LAW_AEGL. AGENT_SETUP initializes a toxic agent into the EAGLE system based on a single EPA table given as a 5 x 3 array of real density values (units mg/m^3). These routines share the required agent data via a named common block, *Agent_AEGLS*. The main program, EAGLE_DRIVER, is a test program illustrating the use of EAGLE and showing one way to specify the AEGL tables for use in AGENT_SETUP. EAGLE_DRIVER is reproduced as Appendix B.

```
c  Common block for agent AEGLs, used for efficiency by function TL_RATE
      Real   Brho(0:6,3), alpha(6,3), rhomax(3), rhomin(3)
      Real   Atime(5), Btime(0:6,3)
      Common /Agent_AEGLS/ Brho, alpha, rhomax, rhomin, Atime, Btime
      Save   /Agent_AEGLS/
```

To initialize EAGLE for a particular agent, call subroutine AGENT_SETUP (*Agent, Arho_xx, taumin, taumax*) to fill the common-block variables with information relevant to the agent to be considered. The three-character string *Agent* labels this agent (e.g. *Agent* = 'NH3'). *taumin* and *taumax* (units seconds) are the minimum onset time allowed and the maximum onset time considered before the accumulation rate is set to hard zero. A data statement in the main

program initializes the five standard AEGL exposure times, given in seconds, in the common array *Atime(5).* In principle a knowledgeable user could change these exposure times if different data were available.

Data statements in the main program also give the 5 x 3 EPA tabulated data, converted from parts per million (PPM) to mg/m^3, for three agents. These agents are used here as examples as they illustrate important features and nonlinearity common to many agents. Whichever *Arho_xx* AEGL array is actually submitted to AGENT_SETUP must also be dimensioned 5 x 3, as on the EPA web site. There are three example choices provided in the test program, *Arho_CL2* for gaseous chlorine, *Arho_NH3* for gaseous ammonia, and *Arho_PLC*, a placeholder array for a mathematically ideal power-law contaminant, determined by subroutine POWER_LAW_AEGL as described below.

The common block actually communicates a 7 by 3 real array of AEGL densities, *Brho(b,k),* and a 6 by 3 real array of power-law exponents, *alpha(b,k),* between the routines. AGENT_SETUP computes and initializes these quantities. The expanded arrays contain added data, based on given user-specified values of *taumin* and *taumax*, to ensure continuity of the toxic loading rates computed for densities falling outside the range given in the table. The variables *taumin* and *taumax* are two real times in seconds specifying the shortest exposure time over which an AEGL 1, 2 or 3 onset can be reached and a time determining the slowest rate of toxic load accumulation before TL_RATE is set to hard zero. We are using taumin = 200 seconds, three times shorter than 10 minutes, and taumax = 24 hours, three times longer than 8 hours.

SETUP_AGENT prints a diagnostic block of output each time it is called. This block records the values that are actually being used in a given run and indicates how the solution is being interpreted. *Atime(b),* b = 1 to 5 are the AEGL exposure times. *Arho(b,k)* are the agent densities that would lead to onset of the AEGL k symptoms after an exposure of *Atime(b)* seconds. *Alpha(b,k)* are derived power law exponents for interpolating toxic loading rates between the density values given as *Arho(b-1,k)* and *Arho(b,k).* The line labeled 'extrap.'above b = 1 in Table 3 records the derived densities and power laws for the high-density extrapolation from the 10-minute AEGLs down to 200 seconds (*taumin*). The corresponding power law is assumed to be identical to *alpha(b,1).*

AGENT_SETUP: agent ID is CL2

	Atime(k)	Brho(b,1)	alpha(b,1)	Brho(b,2)	alpha(b,2)	Brho(b,3)	alpha(b,3)
extrap.	200.0	1.4700	0.0000	8.2600	0.0000	263.3929	1.8948
b = 1	600.0	1.4700	0.0000	8.2600	0.0000	147.5000	1.8948
b = 2	1800.0	1.4700	0.0000	8.2600	0.0000	82.6000	1.8948
b = 3	3600.0	1.4700	0.0000	5.9000	8.5902	59.0000	2.0600
b = 4	14400.0	1.4700	0.0000	2.9500	2.0000	29.5000	2.0000
b = 5	28800.0	1.4700	0.0000	2.0800	1.9836	20.8000	1.9836
extrap.	86400.0	1.4700	0.0000	1.1955	1.9836	11.9545	1.9836

Table 3. Diagnostic printout occurs each time AGENT_SETUP is called. This example is for chlorine (Agent = 'CL2' is a 3 character argument in the calling sequence). For AEGL 1 the alpha values are all 0.0, indicating this is a threshold density for the rapid onset of symptoms (e.g. eyes watering) that is independent of the exposure time. AEGL 3 exposure times follow an inverse quadratic power law. Doubling the concentration reduces the time to AEGL 3 onset by a factor of four. AEGL 2 values show threshold behavior for high densities at short exposure times and an inverse quadratic law for densities lower than 8.26 mg/m^3.

The line labeled 'extrap.' at the bottom of the table gives the derived density and power law for the low-density extrapolation from the 8-hour AEGLs out to 24 hours (*taumax*). The *b = 6*

power law is set to *alpha(b,5)* as shown in the table. The yellow background in Tables 3, 4, and 5 indicates values subject to these extrapolations. The user is free to set *taumin* = 599 sec and *taumax* = 28801 sec, effectively removing these extrapolations. This also says that no matter how dense the chlorine gas, it will take at least 10 minutes to reach a lethal dose.

Figure 1 below plots the continuous toxic loading times for chlorine corresponding to Table 3. At *b = 3* in Table 3 above the power law for AEGL 2 (*alpha(3,2)*) is 8.5902, much greater than 2. This occurs to accommodate the fact that the AEGL 2 symptom onset time will be 200 sec, not 1800 sec, at and above densities of 8.26 mg/m^3 because of the indicated threshold behavior. Therefore the toxic loading rate must increase from 1/3600 sec^{-1} at 5.9 mg/m^3 to 1/200 sec^{-1} at 8.26 mg/m^3, necessitating stronger density dependence. These adjustments are performed automatically by subroutine AGENT_SETUP from the structure of the input table. These algorithms assume that an agent's toxicity monotonically increases, or at least stays constant, as the agent density increases. The adjustments performed by AGENT_SETUP do not stem from additional toxicological information but rather ensure logical consistency.

AGENT_SETUP: agent ID is NH3

	Atime(k)	Brho(b,1)	alpha(b,1)	Brho(b,2)	alpha(b,2)	Brho(b,3)	alpha(b,3)
extrap.	200.0	21.3000	0.0000	156.0000	0.0000	3229.9814	2.0974
b = 1	600.0	21.3000	0.0000	156.0000	0.0000	1913.0000	2.0974
b = 2	1800.0	21.3000	0.0000	156.0000	0.0000	1133.0000	2.0974
b = 3	3600.0	21.3000	0.0000	113.0000	8.9633	779.0000	1.8503
b = 4	14400.0	21.3000	0.0000	77.9000	3.7270	390.0000	2.0037
b = 5	28800.0	21.3000	0.0000	77.9000	0.0000	276.0000	2.0048
extrap.	86400.0	21.3000	0.0000	77.9000	0.0000	159.5579	2.0048

Table 4. Time-dependent AEGL toxic load integrals for ammonia reformatted by EAGLE. AEGL 1 densities are the same for each time, indicating a threshold for the onset of symptoms (e.g. eyes watering) independent of the exposure duration. AEGL 3 concentration values follow a quadratic power law. Doubling the concentration reduces the time to AEGL 3 onset by a factor of four. AEGL 2 values for ammonia show non-power-law behavior with two plateaus (thresholds) and a transition.

In Tables 3 and 4, the quantities *alpha(b,k)* are the power law exponents of AEGL k for the density band from *Brho(b,k)* to the larger density Brho(b-1,k). Thus *alpha(0,k)* (set equal to *alpha(1,k)*) is never used because any density larger than *Brho(0,k)* is mapped to *Brho(0,k)*. Similarly, the power law exponents for b = 6 (labeled "extrap." above) are simply duplicated from the adjacent band b = 5. This exponent is also used to define the lowest density value *Brho(6,k)* that will be considered based on the user-supplied value of *taumax*, here 24 hours (86400 seconds). Any lower density will have its toxic loading rate set to hard zero. Furthermore, the exponent alpha values for a threshold plateau are set to zero and not used. Instead the accumulation rate is set as the inverse of the shortest onset time for that threshold plateau.

Subroutine POWER_LAW_AEGL (*power, Arho, times*) fills a user supplied 5x3 real 'Arho_PLC' array with density values that correspond to the given list of 'times' and the desired power-law exponent, *power* (a real number between zero and ten). This allows a user to study the effects of the nonlinearity in toxic loading for imaginary idealized agents. The densities returned for each AEG Level by POWER_LAW_AEGL are scaled off the 1-hour value so that the same array, Arho_PLC, can be initialized repeatedly for different values of the toxicity power-law exponent. The density for the 1-hour AEGL 2 level is 1.0. The AEGL 1 densities are a factor of 20 lower and the AEGL 3 densities are a factor of 20 higher that the AEGL 2 values. This is clearly evident in Table 5, which captures the test program print out for powers laws of 1.0 (linear dose), 2.0 (quadratic as chlorine), and 3.0 (a stronger nonlinear dependence).

After the array *Arho* has been chosen, filled, or read in and after subroutine AGENT_SETUP has been called, the shared common block of data is ready to be used. Hereafter, any call to Real Function TL_RATE(*rho*, *k*), where k = 1, 2, 3 selects the desired AEGL level, will return the rate, in inverses seconds, at which the toxic load for AEGL k will accumulate for the given value of density. Starting with a toxic load TL of zero, TL is incremented simply by successively accumulating *deltat* * TL_RATE(*rho*, *k*), to evolve the toxic load. A number of examples of this are given in the Fortran test program reproduced as Appendix B, entitled "A Tutorial Driver Program Testing the EAGLE Package"

```
POWER_LAW_AEGL: power: 1.0000
  k = 1, Arho(1-5,1) =  3.0000E-01  1.0000E-01  5.0000E-02  1.2500E-02  6.2500E-03
  k = 2, Arho(1-5,2) =  6.0000E+00  2.0000E+00  1.0000E+00  2.5000E-01  1.2500E-01
  k = 3, Arho(1-5,3) =  1.2000E+02  4.0000E+01  2.0000E+01  5.0000E+00  2.5000E+00

AGENT_SETUP: agent ID is PL1
          Atime(b)   Arho(b,1) alpha(b,1)   Arho(b,2) alpha(b,2)   Arho(b,3) alpha(b,3)
extrap.      200.0       0.9000    1.0000      18.0000    1.0000     360.0001    1.0000
b =  1       600.0       0.3000    1.0000       6.0000    1.0000     120.0000    1.0000
b =  2      1800.0       0.1000    1.0000       2.0000    1.0000      40.0000    1.0000
b =  3      3600.0       0.0500    1.0000       1.0000    1.0000      20.0000    1.0000
b =  4     14400.0       0.0125    1.0000       0.2500    1.0000       5.0000    1.0000
b =  5     28800.0       0.0063    1.0000       0.1250    1.0000       2.5000    1.0000
extrap.    86400.0       0.0021    1.0000       0.0417    1.0000       0.8333    1.0000


POWER_LAW_AEGL: power: 2.0000
  k = 1, Arho(1-5,1) =  1.2247E-01  7.0711E-02  5.0000E-02  2.5000E-02  1.7678E-02
  k = 2, Arho(1-5,2) =  2.4495E+00  1.4142E+00  1.0000E+00  5.0000E-01  3.5355E-01
  k = 3, Arho(1-5,3) =  4.8990E+01  2.8284E+01  2.0000E+01  1.0000E+01  7.0711E+00

AGENT_SETUP: agent ID is PL2
          Atime(b)   Arho(b,1) alpha(b,1)   Arho(b,2) alpha(b,2)   Arho(b,3) alpha(b,3)
extrap.      200.0       0.2121    2.0000       4.2426    2.0000      84.8528    2.0000
b =  1       600.0       0.1225    2.0000       2.4495    2.0000      48.9898    2.0000
b =  2      1800.0       0.0707    2.0000       1.4142    2.0000      28.2843    2.0000
b =  3      3600.0       0.0500    2.0000       1.0000    2.0000      20.0000    2.0000
b =  4     14400.0       0.0250    2.0000       0.5000    2.0000      10.0000    2.0000
b =  5     28800.0       0.0177    2.0000       0.3536    2.0000       7.0711    2.0000
extrap.    86400.0       0.0102    2.0000       0.2041    2.0000       4.0825    2.0000


POWER_LAW_AEGL: power: 3.0000
  k = 1, Arho(1-5,1) =  9.0856E-02  6.2996E-02  5.0000E-02  3.1498E-02  2.5000E-02
  k = 2, Arho(1-5,2) =  1.8171E+00  1.2599E+00  1.0000E+00  6.2996E-01  5.0000E-01
  k = 3, Arho(1-5,3) =  3.6342E+01  2.5198E+01  2.0000E+01  1.2599E+01  1.0000E+01

AGENT_SETUP: agent ID is PL3
          Atime(b)   Arho(b,1) alpha(b,1)   Arho(b,2) alpha(b,2)   Arho(b,3) alpha(b,3)
extrap.      200.0       0.1310    3.0000       2.6207    3.0000      52.4148    3.0000
b =  1       600.0       0.0909    3.0000       1.8171    3.0000      36.3424    3.0000
b =  2      1800.0       0.0630    3.0000       1.2599    3.0000      25.1984    3.0000
b =  3      3600.0       0.0500    3.0000       1.0000    3.0000      20.0000    3.0000
b =  4     14400.0       0.0315    3.0000       0.6300    3.0000      12.5992    3.0000
b =  5     28800.0       0.0250    3.0000       0.5000    3.0000      10.0000    3.0000
extrap.    86400.0       0.0173    3.0000       0.3467    3.0000       6.9336    3.0000
```

Table 5. *Brho(b,k)* and *alpha(b,k)* values for ideal power-law AEGLs with exponent values 1.0 (top), 2.0 (middle), and 3.0 (bottom). AEGL 1 threshold concentrations are the same for each time, indicating a threshold concentration for the onset of symptoms (e.g. eyes watering) independent of the exposure duration. AEGL 3 concentration values follow a quadratic power law. Doubling the concentration reduces the time to AEGL 3 onset by a factor of four. AEGL 2 values for ammonia show a non-power-law behavior with two plateaus.

This program, called EAGLE_DRIVER, is intended as a simple example to use as a template for applying the EAGLE package. Some of the examples can be transported directly in a T&D model to integrate the EPA AEGL toxicity data against any time-dependent toxic agent density profile. The test problems presented are described in the following section.

V. Description of the EAGLE Test Problems

The EAGLE driver program in Appendix B computes a number of results, as formatted tables. Several of these are reproduced here, to illustrate aspects of the EAGLE algorithm and EPA AEGL conditions. EAGLE_DRIVER prints a table of comma separated values (.csv) suitable for plotting that contains the toxic loading times as a function of agent density, *rho*. These times are the inverse of the toxic loading rate, dTL/dt, which is computed by subroutine TL_RATE to integrate Eq. (3). These times, as printed by EAGLE_DRIVER, are plotted in Figure 1. The five tabulated AEGL times are indicated by horizontal black bars. The vertical lines at the low density end of the AEGL 1, 2, and 3 curves indicate densities below which no toxic loading occurs at all.

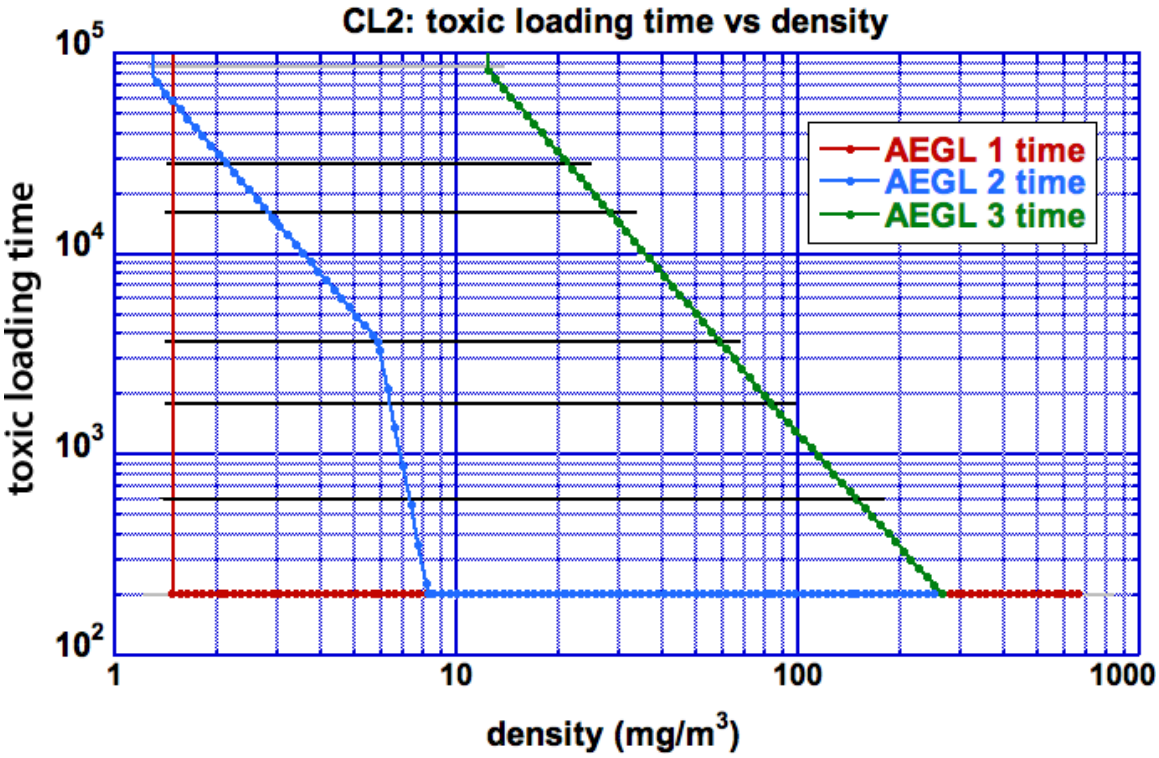

Figure 1. Toxic loading time (seconds = TLrate^{-1}) versus density (mg/m^3) for AEGL 1, 2, and 3 in chlorine. AEGL 1 (red) has a threshold behavior. Density above the threshold signals symptom onset in the shortest time allowed (*taumin* = 200 sec). Below this density (1.47 mg/m^3) the symptoms are not seen. AEGL 2 (blue) shows this threshold behavior for densities above 8.26 mg/m^3 and a quadratic behavior for densities below 5.9 mg/m^3. The intermediate transition region ensures a toxic loading rate that is continuous with the density. AEGL 3 is clearly quadratic throughout the range down through the lowest density specified, 20.8 mg/m^3.

Figure 2 below plots the toxic loading times output by EAGLE_DRIVER for ammonia (NH3) as for chlorine in Figure 1 just above. Again the quadratic nature of the AEGL 3 toxic loading for ammonia over the entire range is evident. This has to have been intended. Here,

however, the AEGL 2 toxic loading time shows a more complicated behavior. AEGL 2 has two threshold plateaus in the EPA Table 2, at a concentration of 220 ppm for specific durations 10 minutes and 30 minutes and at a concentration of 110 ppm for the durations 4 hours and 8 hours. The concentration 160 ppm, tabulated for 60 minutes in Table 2, defines a transition from one threshold behavior to another and is drawn in Figure 2 below, as two different power laws during the transition. Curves with smoothly varying slopes might also have been used, but the sparsity of the data does not encourage such additional complication.

Note that this model also allows a toxic load to accumulate in *taumin* if the density is high enough. This is faster than 20 minutes and is discussed in Section III above.

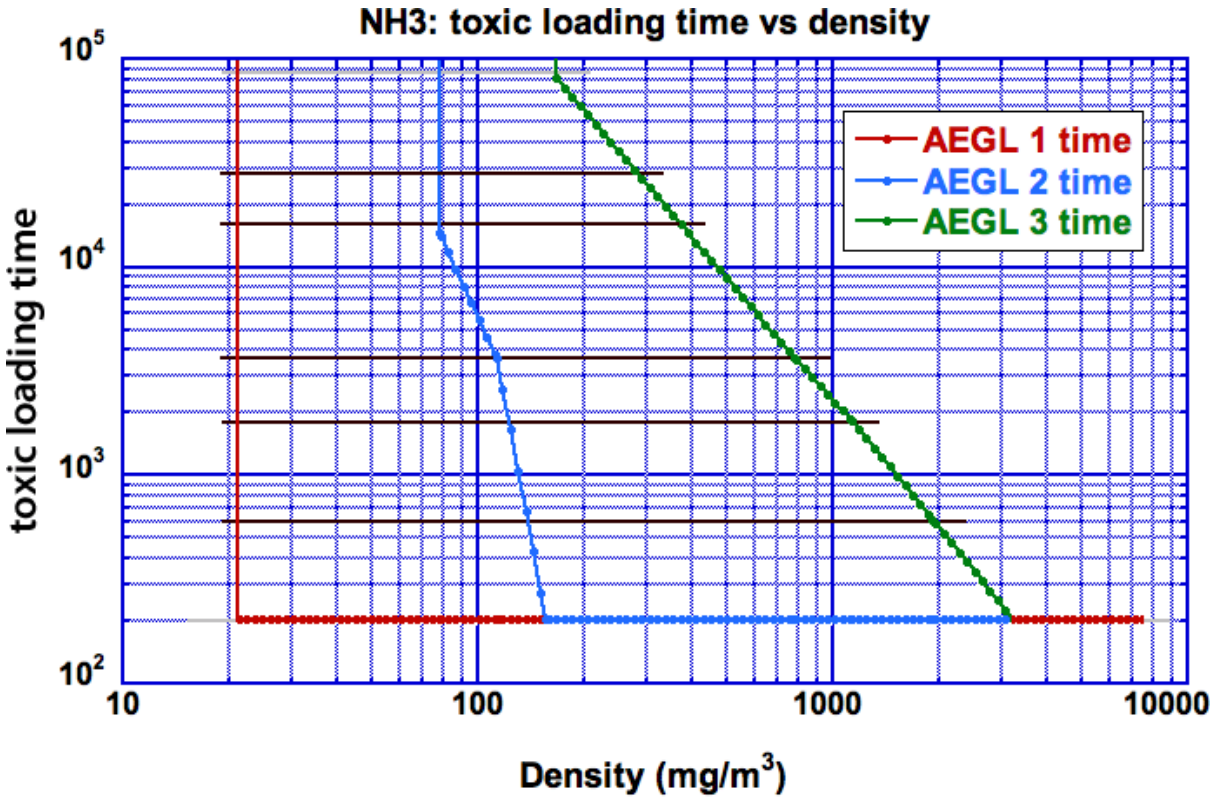

Figure 2. Toxic loading time (seconds = TLrate^{-1}) versus density (mg/m^3) for ammonia: AEGL 1, 2, and 3. AEGL 1 (red) has a threshold behavior across the entire density range with symptom onset at the shortest time allowed (*taumin* = 200 sec). Below this density (21.3 mg/m^3), symptoms are not seen. AEGL 2 (blue) shows two threshold regimes separated by a transition region from 77.9 mg/m^3 to 156 mg/m^3. behavior for densities above 8.26 mg/m^3 and a quadratic behavior for densities below 5.9 mg/m^3. AEGL 3 is clearly quadratic throughout the range down through the lowest density specified, 276 mg/m^3.

Using the values of *taumin* and *taumax* suggested make the toxic loading time, and inversely the TL rate, continuous with density beyond both ends of the EPA table, extending the table by a factor of three in time. Since this numerical model of AEGL toxicity is extrapolated beyond the last table values, we see that death, for example, is still possible for densities somewhat less than 276 mg/m^3 (concentrations less than 390 ppm) it just takes longer. By definition, however, for densities below the extrapolated 24-hour value, 159 mg/m^3 for ammonia, all toxic loading ceases. The nonphysical discontinuity is not eliminated, as we can see, just pushed to longer times. Any user can change these values as they wish including forcing the discontinuity to occur at the EPA table boundaries. Using EAGLE with a number of different input parameters is the way to determine the effect of these assumptions.

Another set of tables that are output by EAGLE_DRIVER provide a consistency check; they demonstrate that integration in time of the values that the function TL_RATE returns actually reproduces the EPA-determined symptom onset times exactly when the EPA table density values are used. Table 6 below, the chlorine example, shows that the integrated toxic load reaches the onset condition $TL_k = 1.0$ for the three AEGLs when the agent exposure density is held constant for each of the five standard exposure durations. The integrated toxic load is printed as a function of the integration time in seconds (left hand column) for a number of integration times out to one hour. For the AEGL 1 table (roughly speaking, eyes watering), TL = 1.0 is reached in 200 seconds in all cases and the toxic load continues increasing at a steady rate thereafter.

AEGL 1 table	1.470	1.470	1.470	1.470	1.470	mg/m**3
time	TL 10 min	TL 30 min	TL 60 min	TL 4 hour	TL 8 hour	
50.0	0.2500	0.2500	0.2500	0.2500	0.2500	
100.0	0.5000	0.5000	0.5000	0.5000	0.5000	
150.0	0.7500	0.7500	0.7500	0.7500	0.7500	
200.0	1.0000	1.0000	1.0000	1.0000	1.0000	
600.0	3.0000	3.0000	3.0000	3.0000	3.0000	
1200.0	6.0000	6.0000	6.0000	6.0000	6.0000	
1800.0	9.0000	9.0000	9.0000	9.0000	9.0000	
2400.0	12.0000	12.0000	12.0000	12.0000	12.0000	
3000.0	15.0000	15.0000	15.0000	15.0000	15.0000	
3600.0	18.0000	18.0000	18.0000	18.0000	18.0000	
AEGL 2 table	8.260	8.260	5.900	2.950	2.080	mg/m**3
time	TL 10 min	TL 30 min	TL 60 min	TL 4 hour	TL 8 hour	
50.0	0.2500	0.2500	0.0139	0.0035	0.0017	
100.0	0.5000	0.5000	0.0278	0.0069	0.0035	
150.0	0.7500	0.7500	0.0417	0.0104	0.0052	
200.0	1.0000	1.0000	0.0556	0.0139	0.0069	
600.0	3.0000	3.0000	0.1667	0.0417	0.0208	
1200.0	6.0000	6.0000	0.3333	0.0833	0.0417	
1800.0	9.0000	9.0000	0.5000	0.1250	0.0625	
2400.0	12.0000	12.0000	0.6667	0.1667	0.0833	
3000.0	15.0000	15.0000	0.8333	0.2083	0.1042	
3600.0	18.0000	18.0000	1.0000	0.2500	0.1250	
AEGL 3 table	147.500	82.600	59.000	29.500	20.800	mg/m**3
time	TL 10 min	TL 30 min	TL 60 min	TL 4 hour	TL 8 hour	
50.0	0.0833	0.0278	0.0139	0.0035	0.0017	
100.0	0.1667	0.0556	0.0278	0.0069	0.0035	
150.0	0.2500	0.0833	0.0417	0.0104	0.0052	
200.0	0.3333	0.1111	0.0556	0.0139	0.0069	
600.0	1.0000	0.3333	0.1667	0.0417	0.0208	
1200.0	2.0000	0.6667	0.3333	0.0833	0.0417	
1800.0	3.0000	1.0000	0.5000	0.1250	0.0625	
2400.0	4.0000	1.3333	0.6667	0.1667	0.0833	
3000.0	5.0000	1.6667	0.8333	0.2083	0.1042	
3600.0	6.0000	2.0000	1.0000	0.2500	0.1250	

Table 6. Time-dependent AEGL toxic load integrals for chlorine accumulated by EAGLE. AEGL 1 threshold concentrations are the same for each time, indicating a threshold concentration for the onset of symptoms (e.g. eyes watering) independent of the exposure duration. AEGL 3 concentration values follow a quadratic power law. Doubling the concentration reduces the time to AEGL 3 onset by a factor of four. AEGL 2 values for ammonia show a non-power-law behavior with two plateaus.

Looking at the bottom line of Table 6 (for AEGL 3) The 10-minute toxic load accumulates to 6.0 in an hour. The 30-minute TL reaches 2.0, and so on. The right-hand column shows the 8-hour AEGL 3 accumulating to only 0.125 in an hour. Accumulating Toxic Load beyond unity has no real meaning. Users have a couple of options beyond simply letting the TL accumulate. TL can simply be capped at unity. Alternately, the time at which TL first exceeds unity can be stored in place of the toxic load in an array of symptom onset times. After some time has elapsed in the simulation, a plot of this onset time array will display graphically how the AEGL onset condition spreads. This is often exactly the information that first responders would like to have from the beginning of an incident.

time,	.1*Rho(t),	CL2 A1,	CL2 A1F,	Rho(t),	CL2 A2,	CL2 A2F,	10xRho(t),	CL2 A3,	CL2 A3F,
0,	0.0000,	0.000,	0.000,	0.0000,	0.000,	0.000,	0.0000,	0.000,	0.000,
50,	0.3535,	0.000,	0.000,	3.5351,	0.003,	0.006,	35.3509,	0.003,	0.004,
100,	0.4995,	0.000,	0.000,	4.9950,	0.010,	0.052,	49.9500,	0.010,	0.014,
150,	0.6103,	0.000,	0.000,	6.1031,	0.023,	0.169,	61.0313,	0.023,	0.036,
200,	0.7015,	0.000,	0.000,	7.0149,	0.060,	0.265,	70.1495,	0.040,	0.057,
250,	0.7784,	0.000,	0.020,	7.7841,	0.162,	0.412,	77.8407,	0.062,	0.094,
300,	0.8433,	0.000,	0.060,	8.4326,	0.376,	0.526,	84.3258,	0.089,	0.126,
350,	0.8970,	0.000,	0.135,	8.9696,	0.626,	0.688,	89.6957,	0.120,	0.177,
400,	0.9398,	0.000,	0.190,	9.3985,	0.876,	0.807,	93.9850,	0.154,	0.218,
450,	0.9721,	0.000,	0.295,	9.7208,	1.126,	0.971,	97.2079,	0.191,	0.279,
500,	0.9938,	0.001,	0.365,	9.9381,	1.376,	1.092,	99.3808,	0.229,	0.325,
550,	1.0053,	0.001,	0.470,	10.0534,	1.626,	1.257,	100.5340,	0.269,	0.391,
600,	1.0072,	0.001,	0.540,	10.0719,	1.876,	1.380,	100.7191,	0.310,	0.439,
650,	1.0001,	0.001,	0.646,	10.0010,	2.126,	1.545,	100.0102,	0.350,	0.506,
700,	0.9850,	0.001,	0.716,	9.8502,	2.376,	1.668,	98.5015,	0.389,	0.553,
750,	0.9630,	0.001,	0.821,	9.6303,	2.626,	1.832,	96.3029,	0.427,	0.616,
800,	0.9353,	0.001,	0.891,	9.3533,	2.876,	1.952,	93.5326,	0.463,	0.659,
850,	0.9031,	0.001,	0.976,	9.0311,	3.126,	2.114,	90.3113,	0.497,	0.715,
900,	0.8676,	0.001,	1.026,	8.6755,	3.376,	2.232,	86.7553,	0.529,	0.753,
950,	0.8297,	0.001,	1.101,	8.2972,	3.626,	2.394,	82.9721,	0.558,	0.801,
1000,	0.7906,	0.001,	1.141,	7.9057,	3.838,	2.507,	79.0569,	0.585,	0.833,
1050,	0.7509,	0.001,	1.186,	7.5091,	3.976,	2.656,	75.0911,	0.609,	0.873,
1100,	0.7114,	0.001,	1.191,	7.1142,	4.064,	2.756,	71.1417,	0.631,	0.899,
1150,	0.6726,	0.001,	1.191,	6.7262,	4.119,	2.898,	67.2617,	0.650,	0.932,
1200,	0.6349,	0.001,	1.191,	6.3492,	4.152,	2.992,	63.4916,	0.667,	0.953,
1250,	0.5986,	0.001,	1.191,	5.9861,	4.173,	3.125,	59.8609,	0.682,	0.979,
1300,	0.5639,	0.001,	1.191,	5.6390,	4.186,	3.207,	56.3896,	0.696,	0.996,
1350,	0.5309,	0.001,	1.191,	5.3090,	4.198,	3.323,	53.0901,	0.708,	1.017,
1400,	0.4997,	0.001,	1.191,	4.9969,	4.209,	3.397,	49.9687,	0.718,	1.030,
1450,	0.4703,	0.001,	1.191,	4.7027,	4.218,	3.493,	47.0269,	0.728,	1.047,
1500,	0.4426,	0.002,	1.191,	4.4263,	4.227,	3.551,	44.2627,	0.736,	1.057,
1550,	0.4167,	0.002,	1.191,	4.1671,	4.234,	3.622,	41.6713,	0.743,	1.070,
1600,	0.3925,	0.002,	1.191,	3.9246,	4.240,	3.655,	39.2465,	0.750,	1.079,
1650,	0.3698,	0.002,	1.191,	3.6980,	4.246,	3.685,	36.9804,	0.756,	1.089,
1700,	0.3486,	0.002,	1.191,	3.4865,	4.251,	3.699,	34.8648,	0.761,	1.095,
1750,	0.3289,	0.002,	1.191,	3.2891,	4.256,	3.711,	32.8908,	0.765,	1.103,
1800,	0.3105,	0.002,	1.192,	3.1050,	4.260,	3.717,	31.0498,	0.769,	1.108,
....									
2400,	0.1652,	0.002,	1.192,	1.6524,	4.286,	3.756,	16.5238,	0.795,	1.147,
3000,	0.0978,	0.003,	1.193,	0.9781,	4.292,	3.768,	9.7808,	0.801,	1.159,
3600,	0.0630,	0.004,	1.193,	0.6295,	4.292,	3.771,	6.2951,	0.802,	1.162,
4200,	0.0432,	0.004,	1.194,	0.4315,	4.293,	3.772,	4.3154,	0.802,	1.163,
4800,	0.0310,	0.005,	1.194,	0.3104,	4.294,	3.773,	3.1043,	0.803,	1.164,
5400,	0.0232,	0.005,	1.195,	0.2319,	4.294,	3.773,	2.3187,	0.804,	1.164,
6000,	0.0178,	0.006,	1.196,	0.1785,	4.295,	3.774,	1.7848,	0.804,	1.165,
6600,	0.0141,	0.007,	1.196,	0.1408,	4.295,	3.774,	1.4080,	0.805,	1.165,
7200,	0.0113,	0.007,	1.197,	0.1134,	4.296,	3.775,	1.1336,	0.805,	1.166,

Table 7. Time-dependent AEGL toxic load integrals for chlorine accumulated by EAGLE. AEGL 1 has a threshold density that is only exceeded when the fluctuation is near its peak. AEGL 3 toxic loading follows a quadratic power law throughout so the fluctuation speeds symptom onset. AEGL 2 has a threshold at high density and a quadratic law for lower densities. In this example the presence of fluctuations actually impedes AEGL 2 onset initially because the density is below threshold much of the time. Later the quadratic toxic loading begins to take over but the density decreases rapidly and toxic loading ceases.

Table 7 above is output extracted from another test that EAGLE_DRIVER performs. The density $\rho(t)$ at a point in the plume point increases and then decreases smoothly in time for two hours, a time-dependence that is provided by a simple analytic function on display in EAGLE_DRIVER. The three AEGL toxic loads for chlorine are then integrated using EAGLE with a time step of 1.0 sec. AEGL 1, as a function of time is found using column 2 as the density for two cases: Column 3, labeled CL2 A1, is AEGL 1 computed from the laminar density 0.1 x $\rho(t)$ (time history in column 2) and column 4, labeled CL2 A1F, uses exactly the same density multiplied by a fluctuation function $[1 + \sin(2\pi\, t / 20)]$ that varies between 0 and 2 and averages 1.0. Columns 5 to 7 contain the same information but for AEGL 2 and using 1.0 $\rho(t)$ as the base density profile. Columns 8 to 10 repeat the computations for AEGL 3 using 10 $\rho(t)$ as the base density profile. The interactions of the fluctuation with the nonlinear toxic loading and with the presence of thresholds for AEGL 1 and for the high-density end of AEGL 2 are clearly evident.

VI. An Application Using the 3D MILES Code FAST3D-CT

To illustrate the use of EAGLE combined with a time-dependent transport and dispersion model, we consider an outdoor case where the density of the contaminant being inhaled arises from a steady release at ground level in realistically fluctuation winds. The contaminant cloud initially spreads from the source location and its density at first increases at points downwind and then, for an acute puff source, decreases in time. Even when the source is steady and emits continuously, natural fluctuations and wind gusts rapidly change the density observed at a point in the plume by about a factor of 3 to 10 above and 10 to 50 below the more slowly changing average concentration. In some situations the people themselves will be moving, either trying to flee or being carried through the plume in a moving vehicle. In others a building's HVAC parameters may be reset during the exposure. In none of these cases is the concentration constant for a long enough period to allow rigorous application of the AEGL guidelines.

Figure 3 below shows a 6 km by 4 km urban region where the building outlines are contoured at ground level. NRL's FAST3D-CT MILES simulation model (Patnaik and Boris, 2005; Patnaik, *et al*, 2007; Boris, *et al.*, 2009, 2010) for this geometry was run for nearly 8 hours of real time at 5-meter spatial resolution with a 3 m/s wind from 300^0 (30^0 north of west). The average velocity and temperature profiles were provided by NRL's COAMPS-OS model (Holt, *et al.*, 2009, 2011) and natural fluctuations at the domain boundaries were imposed on the average profiles using a formulation that has been developed for FAST3D-CT and validated through the OKC field trials and in the University of Hamburg wind tunnel.

After a CFD spin-up interval of 15,000 steps had elapsed (25 minutes), each of the six sources were switched on and continuously emitted thereafter throughout the duration of the run, almost 8 hours real time. The six contaminant densities were recorded every 25 timesteps at ground level across the entire domain for 300,000 timesteps. The resulting data set allows a number of analyses to be performed. Our main interests have been in the extent and distribution and time spectrum of the density fluctuations and in the intermittency of the flow as reflected in the time-varying agent density. The density is low enough that buoyancy effects are not important and we are treating the contaminant as chlorine in this particular test demonstration of the EAGLE package.

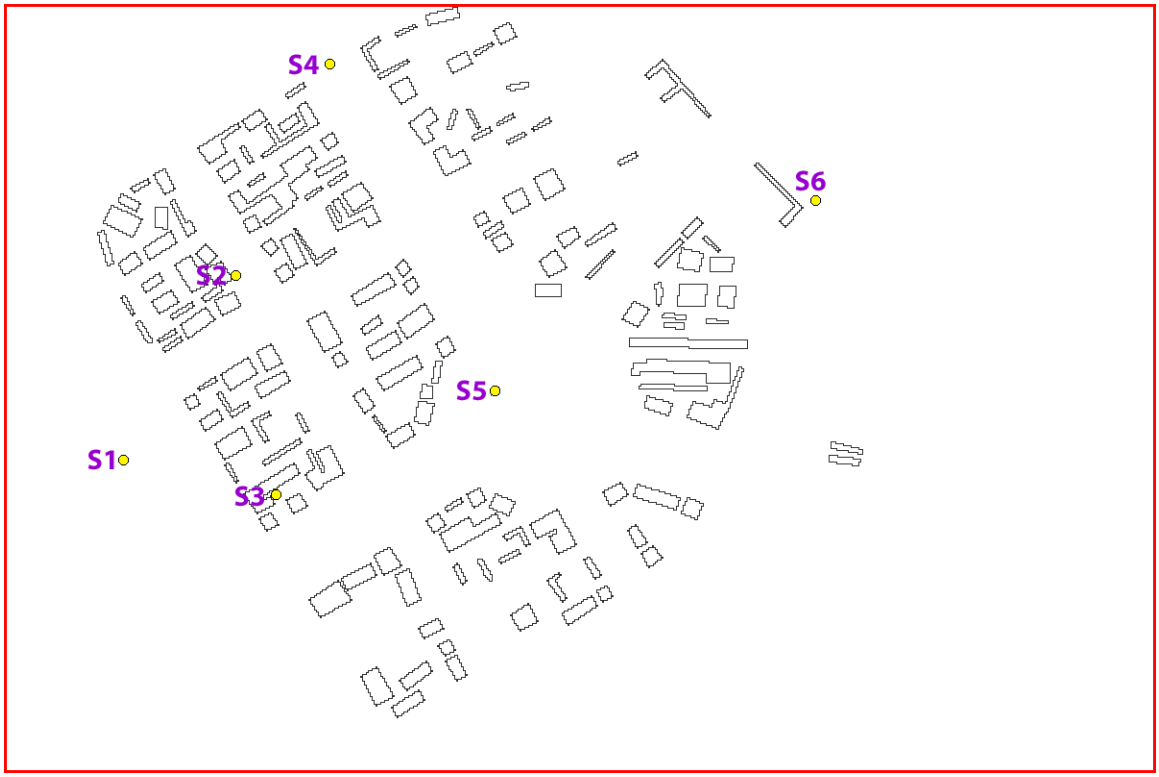

Figure 3. Continuous sources emit agent at 6 locations indicated in a 6 km by 4 km urban domain. The wind is from 300^0 at 3.0 m/s and the temperature is 0^0C. The cell size for the FAST3D-CT simulation was 5 meters. The plumes (e.g. Fig 4 upper left) fluctuate widely in the self-consistent time-dependent winds.

On the left in Fig. 4 below we show an instantaneous color contour plot of the ground-level chlorine density computed by FAST3D-CT 2 hours after the continuous source S4 begin emitting. The source location is marked with the yellow circle at the upper left end of the lavender high-density region, which is just to the right of the label "Source 4". The building cross-sections at ground level appear white in the figures (assuming that no chlorine penetrates). FAST3D-CT is a time-dependent Monotone Integrated Large Eddy Simulation (MILES) model with all of the relevant complex-geometry physics including solar heating with shadows. It has been validated extensively with wind tunnel data and comparisons with field trials in Los Angeles, Oklahoma City, New York City and Hamburg Germany.

Every 2.5 seconds the computed ground-level density was saved and summed to compute the running density average, shown at 2 hours on the right in Fig. 4. The time-averaged densities are smoother, as can be seen, and appear to extend beyond the bounds of the instantaneous plume in some locations. This correct running average is presented to approximate what Reynolds-Averaged Navier-Stokes (RANS) or other steady-state/time-averaging/ensemble models might predict the density to be. The density differences between instantaneous and time-averaged densities do not appear to be great when viewed with a logarithmic color map as in Fig. 4, but their effect on the AEGL regions computed is the two density approximations is important.

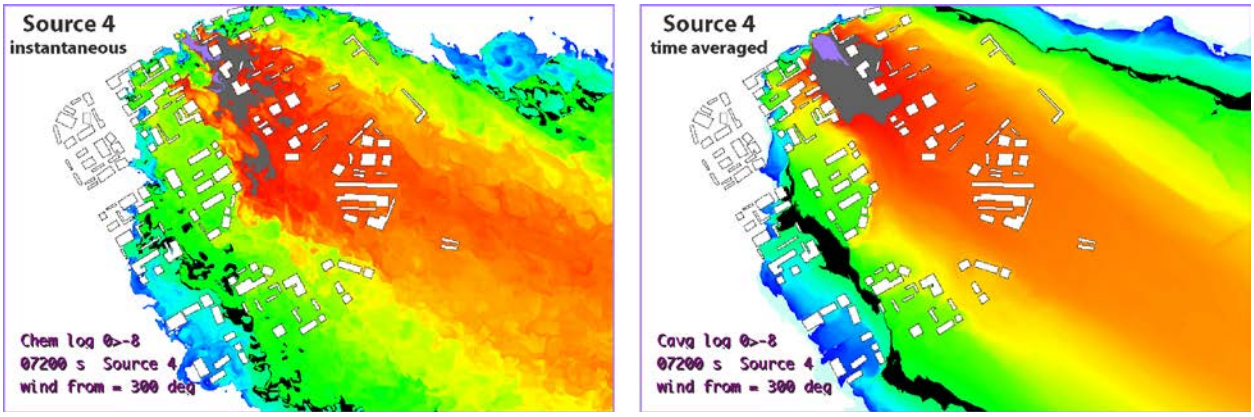

Figure 4. Instantaneous plume density (left) and time averaged plume density (right) for continuous release of chlorine from Source 4 located in the upper left of a 6 km by 4 km domain. Density is shown on a logarithmic scale with black bands indicating concentration values near 0.05 ppm.

Figure 5 below compares the AEGL exposure hazard area predictions, based on the FAST3D-CT time-varying plume, using the new time-dependent AEGL integration routines in the EAGLE package. The two panels below show the AEGL 1 (yellow), AEGL 2 (red), and AEGL 3 (black) hazard areas with and without the natural density fluctuations that are computed in 3D detail by the FAST3D-CT simulation of the fluctuating and gusting winds through the urban geometry. The figure also lists the predicted hazard areas in square kilometers for the two contrasting cases. The area ratios are written on the right hand panel for comparison. Remember from above that AEGL 1 has a threshold for onset at 1.47 mg/m3, and AEGL 2 has a threshold behavior at high density and an exponent 2 power law behavior at lower density, and AEGL 3 is a power law with exponent 2.

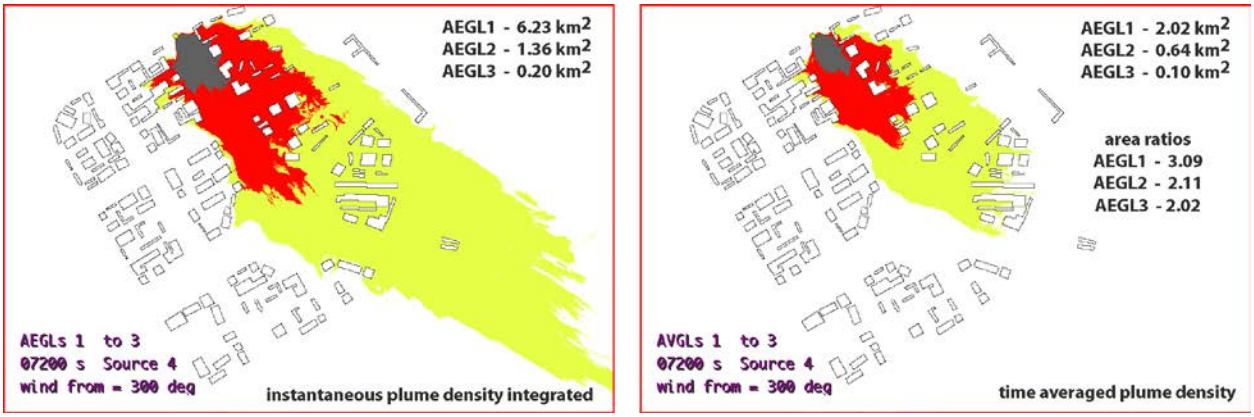

Figure 5. AEGL 1 (yellow), AEGL 2 (red), and AEGL 3 (gray) exposure hazard areas at 2 hours after release computed using the new generalized AEGL routines to integrate the actual, ground level, time-varying densities (left) and the corresponding time-averaged density profiles (right). Natural fluctuations in the density (left), interacting with the non-linear toxic-load behavior of the EPA AEGL tables, increase the hazard areas appreciably.

Including realistic agent density fluctuations increases the hazard area even when the other conditions are all the same. This is not unexpected since it has been recognized for some time that toxic load accumulates much faster than linearly in higher-density regions for many agents. This is further corroborated in Fig. 5 by the fact that the AEGL 1 area ratio, which is based on a limiting threshold of 0.5 ppm, is larger than the AEGL 2 and AEGL 3 ratios that result from a nearly quadratic dependence of accumulation rate with concentration. Though the tables list a

threshold power law exponent as zero, the exponent used is actually infinity as the threshold time varies greatly with no apparent change in the density. In EAGLE, the toxic load accumulation for these cases is evaluating without using the power law.

These results shown in Figs. 4 and 5 are typical of the other five sources calculated. One can expect the chlorine hazard areas for AEGL 2 and AEGL 3 to be up to two or more times larger than a time-averaged or steady-state model might predict. Because AEGL 1 is defined as a threshold for chlorine, the effective power law is infinite and thus area ratios in excess of 2 or 3 are sometimes seen, as estimated by Bogen and Gouveia (2008). When the computed exposure hazard areas begin to extend beyond the computational domain, as would happen with a slightly stronger source in Fig. 3, the contribution to the hazard area cannot be computed everywhere and therefore the areas computed are too small. Since the actual hazard area, computed with fluctuations, extends beyond the FAST3D-CT simulation domain before the averaged-density hazard area does, the correct, overall area ratios are still larger than what we are able to compute and have shown above. Figure 6 below summarizes the results of the exposure hazard area ratios as a function of the toxic load power law exponent.

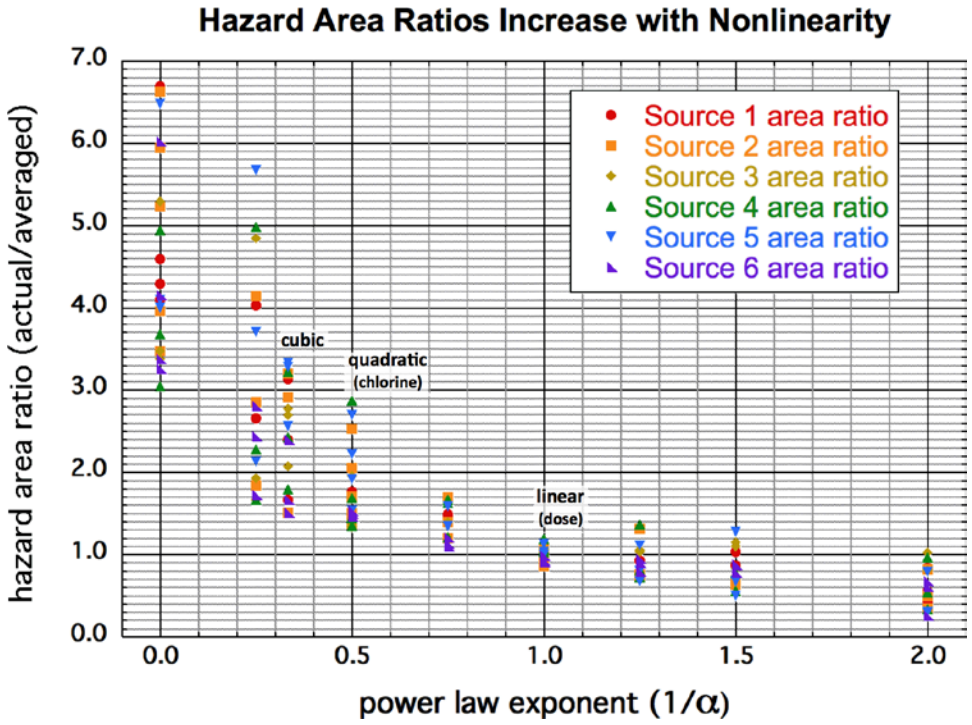

Figure 6. AEGL exposure area ratios for all six sources and a range of power-law exponents. The threshold hazard area behavior, shown at 1/a = 0 near the left edge of the figure, may even extend above an area ratio of 7:1 in some cases.

Many of the EPA AEGL tables for agents have simple power law dependences implicit in them. Furthermore, the limiting case of a single threshold density for symptom onset, which is independent of the length of exposure (e.g. AEGL 1 for chlorine), also corresponds to a power law (exponent of 0.0 or infinity depending on the definition). An ideal power law agent was defined where the exponent can be varied from zero (threshold for symptom onset) to much greater than unity. Figure 6 shows that the hazard area for agents with a toxicity threshold can be 3 to 7 times larger when natural fluctuations are properly included.

There is a lot of work that can still be done. Other FAST3D-CT data sets are can be computed and made available to the community to be subsequently used, perhaps with density

scaling factors, to integrate many different agents and scaled for different source strengths and wind speeds. Such studies should be extended for chlorine other common chemicals and should also include instantaneous and short duration sources. In such acute scenarios, the hazard areas will be even more controlled by the time dependence and the variability caused by different wind field realizations will be significant. Furthermore, the shortcomings of the AEGL definitions for specific time periods will be even more important. For example, Germany must use the 4-hour AEGLs even in situations, for example during the first hour of an accident, where the results give gross over estimates of the current hazard area. The new formulation, coupled with its speed, allows prediction of the evolving hazard areas before the agent arrives. To take full advantage, the ConOps for using this capability in an emergency must therefore be changed accordingly.

VII. Summary

This report presents and describes a simple, local method for applying the EPA Acute Exposure Guideline Levels (AEGLs) to time-varying toxic agent densities such as generated by a transport and dispersion model, by wind tunnel measurements, or in field trials. By interpolating the given EPA tables using the accumulating toxic load as an "induction parameter," the need to consider particular fixed duration exposures at constant density is removed from consideration. There is only one AEGL 1, AEGL 2, and AEGL 3 onset time to be determined at each point in each scenario, not five. The software package, called EAGLE, which implements this algorithm, is structured to recover the EPA-tabulated onset times when the density values, specified on the EPA web site, are held constant for the specified durations. The Fortran code for EAGLE and its test program EAGLE_DRIVER is reproduced as Appendices A and B. This report also provides sample output to ensure a user's installation or method of application is correct.

EAGLE is entirely independent of the method and algorithms used to estimate the time- and space-varying toxic agent density. It can be used with any method including experimental data. This induction parameter approach is very efficient in both computer time and computer memory and thus can be evaluated as any T&D simulation proceeds. Each AEGL being considered requires only one word of memory for each location being tracked. This is adequate to construct plots of AEGL symptom onset time over the entire computational domain as computations are performed without the need for post processing. The software, which has a tiny computational footprint, can also compute the evolving AEGLs for specific locations where sensors have been placed, either in a laboratory, in a city, or on a moving platform.

EAGLE was designed in response to joint project with the University of Hamburg (e.g. Boris, *et al.*, 2009) where software for real-time estimates of predicted health effects was needed in an emergency-management context. The method now in EAGLE removes computational discontinuities and interpretational ambiguities where possible and has been made efficient enough to evaluate the impact of realistic fluctuations in agent concentrations on toxic hazard assessment (e.g. Bogen and Gouveia, 2008 and Stage, 2004). Therefore this package is offered up as a way to focus research on health effects, for use by all levels of users in specific applications, and as an easy-to-use, low-impact metric for comparing models in terms that first-responders and emergency managers will understand and appreciate.

References

Bogen, K.T. and F.J. Gouveia (2008), Impact of spatiotemporal fluctuations in airborne chemical concentration on toxic hazard assessment, *Journal of Hazardous Material*, Volume 152, Issue 1, 21 March 2008, Pages 228–240.

Boris, J., M. Picone, E. Oran, and T. Young Jr. (1980), The Development of an Induction Parameter Model for Simulations of Detonations, Eastern Section: The Combustion Institute, Fall Technical Meetings, Princeton University, 12–14 November 1980.

Boris, J., G. Patnaik, M-Y. Lee, T. Young, B. Leitl, F. Harms and M. Schatzmann (2009), Validation of an LES Urban Aerodynamics Model for Homeland Security, 47th AIAA Aerospace Sciences Meeting, AIAA Paper 2009–1633, Orlando FL, 5–8 January 2009, (AIAA, Reston VA, 2009).

Boris, Jay, Gopal Patnaik, and Keith Obenschain, John Delaney, and John Donnelly (2010), Fast and Accurate Prediction of Windborne Contaminant Plumes for Civil Defense in Cities, Proceedings: Computational Wind Engineering Conference 2010 (CWE'2010), 23–27 May 2010, Chapel Hill, NC.

Holt, T., J. Pullen, and C. H. Bishop (2009), Urban and ocean ensembles for improved meteorological and dispersion modeling of the coastal zone. *Tellus*, **61A**, 232-249.

Holt, T., J. Cummings, C. Bishop, J. Doyle, X. Hong, S. Chen, and Y. Jin, (2011), Development and testing of a Coupled Ocean-Atmosphere Mesoscale ensemble Prediction System COAMPS-OS). *Ocean Dynamics*, **61**, 1937-1954.

Oran, E.S and Boris, J.P. (2001), **Numerical Simulation of Reactive Flow: 5–5.1,** The Induction Parameter Model, pp 150–151, (2nd edition, Cambridge University Press, New York NY, 2001).

Patnaik, G. and J.P. Boris (2005), Fast and Accurate CBR Defense: Bringing HPC to the Warfighter and First Responder, AIAA Paper 2005-7166, 1st AIAA InfoTech@Aerospace Meeting, Arlington VA, 26–29 September 2005.

Patnaik, Gopal, Fernando Grinstein, Jay Boris, Theodore Young, and Oskar Parmhed (2007), Large Scale Urban Simulations, Chapter 17, pp 502–530, in **Implicit Large Eddy Simulation: Computing Turbulent Flows**, F. Grinstein, L. Margolin, and W. Rider (eds), Cambridge University Press, 2007.

Stage, Steven A. (2004), Determination of Acute Exposure Guidelines in a Dispersion Model, *Journal of the Air and Waste Management Association* **54**: 49–59, ISSN 1047-3289, Jan 2004.

Ten Berge, W.F., Zwart, A. Appelman, L.M. (1966), Concentration-Time Mortality Response Relationship of Irritant and Systemically Acting Vapours and Gases, *Journal of Hazardous Materials* **13**:301–309, 1986.

U.S. Environmental Protection Agency AEGL Program (2012), http://www.epa.gov/oppt/aegl

Acknowledgements

The authors wish to thank Keith Obenschain, Mi-Young Obenschain, Lisa Williams, Adam Moses, David Fyfe, and Theodore Young, Jr., the NRL contaminant transport team who have made this technology possible in many ways. ONR, through NRL's Core Research Program, the ONR Naval Expeditionary Dog Program, and Hamburg Germany, through the University of Hamburg, have supported aspects of the work discussed here.

Appendix A: Software for Computing the Time-Dependent AEGL Onsets

```
C  ********************************************************************.*********

      Subroutine POWER_LAW_AEGL ( power, Arho, times )

C  ____________________________________________________________________._________

c  POWER_LAW_AEGL fills the AEGL Arho(5,3) density table with a 'power' law
c  toxicity dependence.  Such a power law is used by EPA for at least portions
c  of the density-dependent toxic loading for many agents and toxic industrial
c  chemicals (TICs).  Arho_PLC is defined in the main program EAGLE.  The
c  formulae below use the 60-minute AEGL density values as fixed and fill in
c  the 30-minute, 10-minute, 4-hour, and 8-hour values using the power laws.

c  power        Desired toxicity power law exponent (typically 0 to 10)
c                   power = 1.0 (usual dose)  power = 2.0 like chlorine, etc)
c  Arho         5 x 3 array into which an analytic EPA table is constructed.
c                   NOTE: AEGL 1&3 densities are .05 & 20 times AEGL 2 values
c  times        (units seconds) Standard onset times used by the EPA

c  Program written initially  Jay P. Boris   9 Nov 2011
c  Most recent modifications  Jay P. Boris   6 Jun 2012

c  This software is distributed by the U.S. Naval Research Laboratory without
c  restriction and without warranty, implied or otherwise.  Do not repackage,
c  transmit, or otherwise re-distribute this software without including this
c  distribution statement.
c  ____________________________________________________________________._________

c  Declare the arguments and associated local variables. . .
c  ____________________________________________________________________._________
      Implicit  NONE

      Real,intent(in)::      power, times(5)
      Real,intent(inout)::   Arho(5,3)   ! This is the constructed AEGL table
      Integer                i, k
c  ____________________________________________________________________._________

c  Scale everything off of Arho(3,k) the 1 hour values, assumed already set
      Do k = 1, 3  ! Loop over the three different levels, AEGL 1, 2, and 3
         Arho(1,k) = Arho(3,k) * ( times(3)/times(1) )**(1.0/power)
         Arho(2,k) = Arho(3,k) * ( times(3)/times(2) )**(1.0/power)
         Arho(4,k) = Arho(3,k) * ( times(3)/times(4) )**(1.0/power)
         Arho(5,k) = Arho(3,k) * ( times(3)/times(5) )**(1.0/power)
      End Do

      Write ( *,1001 ) power, ( ( Arho(i,k), i = 1, 5 ), k = 1, 3 )
 1001 Format ( 1x, /, ' POWER_LAW_AEGL: power:', F7.4, /,
     &   '   k = 1, Arho(1-5,1) =', 1P5E12.4, /,
     &   '   k = 2, Arho(1-5,2) =',   5E12.4, /,
     &   '   k = 3, Arho(1-5,3) =',   5E12.4 )

      Return
      End Subroutine POWER_LAW_AEGL

C  ********************************************************************.*********

      Real Function TL_RATE( rho, k )

C  ____________________________________________________________________._________

c  TL_RATE ('Toxic Load Rate') compute dTL/dt, the rate (units per sec) at which
c  the toxic load accumulates for AEGL "k" ( k = 1, 2, or 3).  When toxic_load
c  accumulates to 1.0 or greater, the corresponding EPA criterion for AEGL k has
c  been reached.  This version is based on a linear interpolation in a
```

```
c  manufactured table.

c  rho          density of agent in mg/m**3 at the current time.
c  k            k = 1, 2, 3 for AEGL 1, AEGL 2, AEGL 3

c  The formula being used, for k = 1, 2, 3 (AEGL) is:

c    TL_RATE = dTL/dt = (rho/Brho(b,k))**alpha(b,k) / Btime(b,k)

c  Here Brho expands the EPA table of symptom onset times for computational
c  efficiency, as determined in subroutine AGENT_SETUP, and alpha is the
c  corresponding array of power law exponents to interpolate the Brho table.
c  b indexes the 6 bands between the specified onset times while allowing for
c  short extrapolations to a taumin somewhat less than 600 seconds and a taumax
c  somewhat greater than 8 hours.

c  Program written initially  Jay P. Boris  31 Oct 2011
c  Most recent modifications  Jay P. Boris  31 Oct 2011

c  This software is distributed by the U.S. Naval Research Laboratory without
c  restriction and without warranty, implied or otherwise.  Do not repackage,
c  transmit, or otherwise re-distribute this software without including this
c  distribution statement.
c _____________________________________________________________________.__________

c  Declare the arguments and associated local variables. . .
c _____________________________________________________________________.__________
      Implicit  NONE

      Integer,intent(in):: k
      Real,intent(in)::    rho
      Integer              b
      Real                 rate_tmp

c  Common block for agent AEGLs, used for efficiency by function TL_RATE
      Real   Brho(0:6,3), alpha(6,3), rhomax(3), rhomin(3)
      Real   Atime(5), Btime(0:6,3)
      Common /Agent_AEGLS/ Brho, alpha, rhomax, rhomin, Atime, Btime
      Save   /Agent_AEGLS/

c _____________________________________________________________________.__________
c  Compute the scaled density raised to the exponent.  Use all AEGL values to
c  interpolate between given times (densities) ...
      If ( rho .lt. Brho(5,k) ) Then   ! Extrapolate with a maximum time
         TL_RATE = 1.0E-6     ! sec-1    no accumulation > infinite time !
         If ( alpha(6,k) .eq. 0.0 ) Return  ! This is a threshold
         If ( rho.ge.rhomin(k) .and. alpha(6,k).ne.0.0 ) Then
            TL_RATE = (rho/Brho(5,k))**alpha(5,k) / Btime(5,k)
         End If
         Return
      End If

c  When rhomax is a threshold, this also works ...
      If ( rho .ge. rhomax(k) ) Then
         TL_RATE = 1.0/Btime(0,k)   ! There is a maximum accumulation rate
         Return
      End If

      If ( rho .lt. Brho(4,k) .and. Brho(4,k).ne.Brho(5,k) ) Then
         b = 5                    ! Lower bound of the density range ...
      Else If ( rho .lt. Brho(3,k) .and.
     &                   Brho(3,k).ne.Brho(4,k) ) Then
         b = 4
      Else If ( rho .lt. Brho(2,k) .and.
     &                   Brho(2,k).ne.Brho(3,k) ) Then
         b = 3
      Else If ( rho .lt. Brho(1,k) ) Then
```

```
         b = 2
      Else If ( rho .lt. rhomax(k) ) Then
         b = 1
      End If

      rate_tmp = (rho/Brho(b,k))**alpha(b,k) / Btime(b,k)
c      Write ( *, 1001 ) rho, k, b, alpha(b,k),
c     &             rate_tmp, 1.0/rate_tmp
 1001 Format ( 'TBL_RATE - rho:', 1PE12.4, '  k,band:', 2I2,
     &   '  alpha(band,k):', F8.5, '  rate_tmp:', F8.5,
     &   '  1/rate_tmp: ', F10.2 )
      TL_RATE = rate_tmp

      Return
      End Function TL_RATE

c  ***********************************************************************.*********

      Subroutine AGENT_SETUP ( Agent, Arho_xx, taumin, taumax )

c  ____________________________________________________________________________._________

c  AGENT_SETUP converts a 5 x 3 EPA AEGL table and stores the result into the
c  named common block /Agent_AEGLS/ for efficient use by function TL_RATE.
c  Auxiliary quantities and power-law interpolants are precomputed and stored.
c  When toxic_load accumulates to 1.0 or greater, for AEGL "k" (1, 2, or 3)
c  onset criterion has been reached.

c  Agent    3 character name of the agent, e.g. 'CL2'
c  Arho_xx      Table densities (for AEGL 1, 2, & 3, interpolation break points
c  alpha    exponent of density dependence of toxic load


c  Arho        Table densities for AEGL 1, 2, & 3, interpolation break points
c  alpha    exponent of density dependence of toxic load
c  tau_min     shortest time (max density) for which tl accumulates



c  The formula being used when rho(b) < rho in 'band' b is:
c  dTL/dt(rho,k)  =  (rho/Brho(b,k))**alpha(b,k) / Btime(b,k)

c  Program written initially  Jay P. Boris  30 Oct 2011
c  Most recent modifications  Jay P. Boris   4 Nov 2011

c  This software is distributed by the U.S. Naval Research Laboratory without
c  restriction and without warranty, implied or otherwise.  Do not repackage,
c  transmit, or otherwise re-distribute this software without including this
c  distribution statement.
c  ____________________________________________________________________________._________

c  Declare the arguments and associated local variables. . .
c  ____________________________________________________________________________._________
      Implicit  NONE

      Character*3,intent(in)::  Agent
      Real,intent(in)::         Arho_xx(5,3), taumin, taumax
      Integer              k, b

c  Common block for agent AEGLs, used for efficiency by function TL_RATE
      Real   Brho(0:6,3), alpha(6,3), rhomax(3), rhomin(3)
      Real   Atime(5), Btime(0:6,3)
      Common /Agent_AEGLS/ Brho, alpha, rhomax, rhomin, Atime, Btime
      Save   /Agent_AEGLS/
c  ____________________________________________________________________________._________

c  Transfer particular agent 'xx' into the common block and compute the rhomax
```

```
c  and alpha values ...
c  taumin should have been set externally, e.g. 200 sec ...
c  taumax should have been set externally, e.g. 24 hours
c _____________________________________________________________________.__________
      Do k = 1, 3
         Do b = 1, 5
            Brho(b,k) = Arho_xx(b,k)
            Btime(b,k) = Atime(b)     ! They start the same for each AEGL
         End Do
         Btime(0,k) = taumin
         Btime(6,k) = taumax
      End Do

      Do k = 1, 3
         Do b = 2, 5   ! Power law trial values (then correct for thresholds)
            If ( Brho(b-1,k) .eq. Brho(b,k) ) Then
               alpha(b,k) = 0.0
            Else
               alpha(b,k) = alog( Atime(b)/Atime(b-1) ) /
     &                     alog( Brho(b-1,k)/Brho(b,k) )
            End If
         End Do
         alpha(1,k) = alpha(2,k)   ! Simple extrapolation to short times
         alpha(6,k) = alpha(5,k)   ! Simple extrapolation to long times
      End Do

c  Extrapolate of the edges of the AEGL table to given taumax, taumin.  For
c  densities larger than rhomax, the rate of TL accumulation is set to
c  1.0/taumin corresponding to the minimum effective exposure time, 'taumin.'
c  For densities lower than rhomin(k), the TL rate is set to hard zero.
c _____________________________________________________________________.__________
      Do k = 1, 3
         If ( alpha(2,k) .eq. 0.0 ) Then    ! Flat => threshold at high density
            rhomax(k) = Brho(1,k)           ! Threshold = Same off the edge
            Brho(0,k) = rhomax(k)
         Else   ! Value of alpha(1,k) already exists => different rho values
            rhomax(k) = Brho(1,k) *
     &                 (Btime(1,k)/taumin)**(1.0/alpha(1,k))
            Brho(0,k) = rhomax(k)           ! Now bigger than Brho(1,k)
c            Write ( *,* ) 'k, rhomax:', k, rhomax(k), Brho(1,k),
c     &         Btime(1,k)/taumin, 1.0/alpha(1,k)
         End If

         If ( Brho(4,k).eq.Brho(5,k) ) Then ! => Threshold at low density
            rhomin(k) = Brho(5,k)           ! Threshold = Same off the edge
            Brho(6,k) = rhomin(k)
         Else   ! When different rhos, a value of alpha(5,k) already calculated
            rhomin(k) = Brho(5,k) *
     &                 (Btime(5,k)/taumax)**(1.0/alpha(5,k))
            Brho(6,k) = rhomin(k)           ! Now less than than Brho(5,k)
         End If
      End Do

c  Correct the times to account for threshold values (alpha(b,k) = 0.0).  This
c  collapses the times to the shortest time for which the threshold applies ...
c _____________________________________________________________________.__________
      Do k = 1, 3
         Do b = 1, 6
            If ( alpha(b,k) .eq. 0.0 ) Btime(b,k) = Btime(b-1,k)
         End Do
c  Diagnostic
c          Write ( *, 2001 ) k, ( Btime(b,k), b = 0, 6 )
c 2001     Format ( 'k=', I2, ' Btime(:,k)', 7F10.2 )
      End Do

c  Now attempt to correct for threshold transitions occuring at higher density.
c  The power law will have to be modified in the band to the right (lower
```

```
c  density) of the threshold by using the shortest time, already set in the
c  Btime array.
c  ____________________________________________________________________.__________
      Do k = 1, 3
         Do b = 2, 4  ! Stay on the range ...
            If ( alpha(b-1,k).eq.0.0 .and. alpha(b,k).gt.0.0 ) Then
               alpha(b,k) = alog( Btime(b,k)/Btime(b-1,k) ) /
     &                     alog( Brho(b-1,k)/Brho(b,k) )
            End If
         End Do
      End Do

c  Print a diagnostic in case needed ...
      Write ( *,* ) '    '
      Write ( *, * ) 'AGENT_SETUP: agent ID is ', Agent
      Write ( *,1004 )
 1004 FORMAT ('          Atime(k)   Brho(b,1) alpha(b,1)   ',
     &            'Brho(b,2) alpha(b,2)   Brho(b,3) alpha(b,3)' )
      Write ( *,1006 ) taumin, ( rhomax(k), alpha(1,k), k = 1, 3 )
      Do b = 1, 5
         Write ( *,1005 ) b, Atime(b), ( Brho(b,k),
     &                          alpha(b,k), k = 1, 3 )
      End Do
      Write ( *,1006 ) taumax, ( rhomin(k), alpha(5,k), k = 1, 3 )
      Write ( *,* ) '    '
 1005 Format ( ' b = ',I2, 3X, F8.1, 3X, 2F10.4, 3X, 2F10.4,
     &         3X, 2F10.4 )
 1006 Format ( ' extrap.', 2X, F8.1, 3X, 2F10.4, 3X, 2F10.4,
     &         3X, 2F10.4 )

      Return
      End Subroutine AGENT_SETUP

c  ********************************************************************.*********
```

Appendix B: A Tutorial Driver Program Testing the EAGLE Package

```
c  ********************************************************************.*********

      Program EAGLE_DRIVER

c  ____________________________________________________________________.__________

c  EAGLE_DRIVER is a tutorial driver program for the EAGLE subroutine package
c  for integrating the EPA AEGL toxicity data against time-dependent toxic agent
c  densities such as generated by atmospheric transport and dispersion models.

c  Program written initially  Jay P. Boris  17 Jul 2011
c  Most recent modifications  Jay P. Boris  14 Jul 2012

c  This software is distributed by the U.S. Naval Research Laboratory without
c  restriction and without warranty, implied or otherwise.  Do not repackage,
c  transmit, or otherwise re-distribute this software without including this
c  distribution statement.
c  ____________________________________________________________________.__________

c  Declare the main program data and associated local variables. . .
c  ____________________________________________________________________.__________
      Implicit  NONE

      Character*3  Agent
      Integer   icase, b, bb, k,kk, m
      Integer   istep, stepdel, stepend
      Real      PI, delta_t, time, fluct, ScaleF, SF
```

```
      Real      Arho_NH3(5,3), Arho_CL2(5,3), Arho_PLC(5,3), power
      Real      TLoad(0:14400,3,2)  ! 7200 seconds, 3 AEGLS, 2 cases = 6 curves
      Real      Rho0(0:14400), RhoF(0:14400), rhotemp, rhosave(100)
      Real      dtl_dt(3), TL_RATE, toxic_load(5,3), density(3)
      Real      bigrho, smlrho, rhotest, rhoprev, taumin, taumax

      Data      PI / 3.14159265 /
      Data      Atime / 600., 1800., 3600., 14400., 28800. /  ! AEGL durations

c  Formula at 20-deg C:  mg/m**3 = ppm * (GMW/24.04) = .7084 for NH3
      Data     Arho_NH3 /  21.3,  21.3,  21.3,  21.3,  21.3,  ! @ 20 deg C
     &                     156.,  156.,  113.,  77.9,  77.9,
     &                    1913., 1133.,  779.,  390.,  276. /

c  Formula at 20-deg C:  mg/m**3 = ppm * (GMW/24.04) = 2.95 for CL2
      Data     Arho_CL2 / 1.47,  1.47,  1.47,  1.47,  1.47,   ! @ 20 deg C
     &                     8.26,  8.26,  5.90,  2.95,  2.08,
     &                   147.5,  82.6,  59.0,  29.5,  20.8 /

      Data     Arho_PLC / 0.05,  0.05,  0.05,  0.05,  0.05,   !  artificial  !
     &                    1.00,  1.00,  1.00,  1.00,  1.00,
     &                    20.0,  20.0,  20.0,  20.0,  20.0 /  ! factor of 20 !

c  Common block for agent AEGLs, used for efficiency by function TL_RATE
c _____________________________________________________________________._________
      Real   Brho(0:6,3), alpha(6,3), rhomax(3), rhomin(3)
      Real   Atime(5), Btime(0:6,3)
      Common /Agent_AEGLS/ Brho, alpha, rhomax, rhomin, Atime, Btime
      Save   /Agent_AEGLS/
c _____________________________________________________________________._________

c  Initialize some variables before filling common block 'Agent_AEGLS.' 'taumin'
c  and 'taumax' are user determined.  To remove the extrapolation bands that
c  extend the EPA AEGL table to slightly higher densities (shorter times) and to
c  slightly lower densities (longer times), set taumin = 599.0 and set
c  taumax = 3600.0*8.0 + 1.0 seconds.
c _____________________________________________________________________._________
      taumin = 200.0        ! No AEGL k is accumulated in less than 200 sec
      taumax = 3600.0*24.0  ! 24 hours maximum accumulation time
      ScaleF = 1.0          ! The plume densities are not scaled ...
      Agent = 'NH3'
      Agent = 'PLm'         ! Any power from 0.0 to 9.0
      Agent = 'CL2'

c  ********************************************************************.*********
c  Print out a simple test case (reproduce chlorine table values exactly ...)
c  ********************************************************************.*********
c      Agent = 'NH3'        ! Alternate test
c      Call AGENT_SETUP ( Agent, Arho_NH3, taumin, taumax )
      Agent = 'CL2'
      Call AGENT_SETUP ( 'CL2', Arho_CL2, taumin, taumax )

      Do b = 1, 5
         Do k = 1, 3
            dtl_dt(k) = TL_RATE( Brho(b,k), k )
         End do
         Write ( *, 1003 ) b, ( Brho(b,k),
     &      alpha(b,k), 1.0/max(dtl_dt(k),1.0E-8), k = 1,3 )
      End Do
      Write ( *,* ) '    '  ! Space between tests ...

c  ********************************************************************.*********
c  Calculate an extended test all the way from rhomax to rhomin.
c  ********************************************************************.*********
      bigrho = max( rhomax(1), rhomax(2), rhomax(3) )
      smlrho = min( rhomin(1), rhomin(2), rhomin(3) )
```

```
      Write ( *,* ) Agent, '  bigrho:', bigrho, '  smlrho:', smlrho
      Write ( *, 1005)
      Do bb = 0, 100
         rhotest = smlrho*(bigrho/smlrho)**(0.01*real(bb))
         If ( bb .gt. 0 ) Then      ! Check for all tabled densities ...
            rhosave(bb) = rhotest   ! Keep for honest timing test ...
            rhoprev = smlrho*(bigrho/smlrho)**(0.01*real(bb-1))
            Do k = 1, 3
            Do b = 1, 5
               If ( rhoprev.lt.Brho(b,k) .and.
     &                rhotest.gt.Brho(b,k) ) Then  ! Important density missed
                  rhotemp = Brho(b,k)               ! Temporary test density
                  Do kk = 1, 3
                     dtl_dt(kk) = TL_RATE( rhotemp, kk )
                  End do
                  Write ( *, 1004 ) -bb, rhotemp,
     &                 ( 1.0/dtl_dt(kk), dtl_dt(kk), kk = 1,3 )
               End If  ! This density value is from the table ...
            End Do
            End Do
         End if

c  Now compute the toxlic loading rate for the rhotest density ...
         Do k = 1, 3
            dtl_dt(k) = TL_RATE( rhotest, k )
         End do
         Write ( *, 1004 ) bb, rhotest,
     &      ( 1.0/dtl_dt(k), dtl_dt(k), k = 1,3 )
      End Do
      Write ( *,* ) '    '  ! Space between tests ...

 1005 Format ( ' index,   density,   time AEGL1, rate AEGL1,  ',
     &   '   time AEGL2, rate AEGL2,     time AEGL3, rate AEGL3,' )
 1004 Format ( I5,',', F11.5,',', F13.3,',', F11.7,',   ',
     &         F13.3,',', F11.7,',   ', F13.3,',', F11.7,',' )
 1003 Format ( I2, '  Arho1,alpha1,t1:', F9.4, F7.4, F8.1,
     &             '  Arho2,alpha2,t2:', F9.4, F7.4, F8.1,
     &             '  Arho3,alpha3,t3:', F9.4, F7.4, F8.1 )

c  *****************************************************************.*********
c  Perform a short timing test of 300 million calls. Takes ~6-7 seconds on 1
c  processor of a Macintosh.
c  *****************************************************************.*********
      Write ( *,* ) ' Begin  timing test ...'
      Do k = 1, 1000000    ! Repeat test over and over.
         Do bb = 1, 100  ! Saved density values
            dtl_dt(1) = TL_RATE( rhosave(bb), 1 )
            dtl_dt(2) = TL_RATE( rhosave(bb), 2 )
            dtl_dt(3) = TL_RATE( rhosave(bb), 3 )
         End Do
        If ( mod(k,100000) .eq. 0 ) Write ( *,* ) k*300, ' calls '
      End Do
      Write ( *,* ) ' Finish timing test ...'

c  *****************************************************************.*********
c  Now integrate for the specified durations and check chlorine accumulation ..
c  *****************************************************************.*********
      stepend = 3600
      stepdel = 50     ! 5/6 minute
      delta_t = real(stepdel)
      Call AGENT_SETUP ( 'CL2', Arho_CL2, taumin, taumax )
      toxic_load = 0.0  ! build up starts at zero, 1.0 = final condition.

c  Print out the tabled density values ...
 3000    Format ( 'AEGL ', I1,' table', 5F11.3, ' mg/m**3' )
 3011 Format ( '    time       TL 10 min  TL 30 min  TL 60 min ',
     &   ' TL 4 hour  TL 8 hour ' )
```

```
 3012 Format ( '    ----       ---------  ---------  --------- ',
     &   ' ---------  --------- ' )

      Do k = 1, 3
         Write ( *,* ) '    '
         Write ( *, 3000 ) k, ( Brho(b,k), b=1,5 )
         Write ( *,* ) '    '
         Write ( *,3011 )
         Write ( *,3012 )
      Do istep = 1, stepend/stepdel
         time = real(istep) * stepdel

         Do b = 1, 5   ! Chose which AEGL time is being checked ...
            density(:) = Brho(b,:)

c  Calculate the rate of toxic load accumulation for a given density (mg/m**3)
c  and integrate the toxic load for each of the 15 test cases ...
            toxic_load(b,k) = toxic_load(b,k)
     &              + TL_RATE( density(k), k ) * delta_t
         End do

         If ( istep .lt. 5 .or. mod(istep,12) .eq.0 )
     &      Write ( *, 3001 ) time, ( toxic_load(b,k), b=1,5 )
      End Do ! Loop over the AEGL 1, 2, or 3
 3001 Format ( F8.1, 4X, 5F11.4 )

      End Do

c  ********************************************************************.*********
c  An analytic time history is established and run three times, once for CL2,
c  once for ammonia, and once for PLC + 1.0, the linear dose case.  This same
c  time history is modified by a fast oscillating component mimicing the fluc-
c  tuating agent density from gusts.
c  ********************************************************************.*********
      TLoad = 0.0
      Do m = -1, 3
         If ( m .eq. -1 ) Then  ! chlorine ...
            Call AGENT_SETUP ( 'CL2', Arho_CL2, taumin, taumax )
            ScaleF = 0.5
         Else If ( m .eq. 0 ) Then   ! ammonia ...
            Call AGENT_SETUP ( 'NH3', Arho_NH3, taumin, taumax )
            ScaleF = 8.0
         Else If ( m .gt. 0 ) Then
            power = min( real(m), 9.0 )   ! Arbitrary 1-digit limit of power
            Write (Agent, 3999) m         ! Agent will be PLm
            Call POWER_LAW_AEGL ( power, Arho_PLC, Atime )
            Call AGENT_SETUP ( Agent, Arho_PLC, taumin, taumax )
            ScaleF = 0.2
         End If
 3999    Format ( 'PL',I1.1 )

 4002    Format ( ' time, .1*Rho(t),  CL2 A1, CL2 A1F, ',
     &                '    Rho(t),  CL2 A2, CL2 A2F, ',
     &                ' 10xRho(t),  CL2 A3, CL2 A3F,' )
 4003    Format ( ' time, .1*Rho(t),  NH3 A1, NH3 A1F, ',
     &                '    Rho(t),  NH3 A2, NH3 A2F, ',
     &                ' 10xRho(t),  NH3 A3, NH3 A3F,' )
 4004    Format ( ' time, .1*Rho(t),  PL# A1, PL# A1F, ',
     &                '    Rho(t),  PL# A2, PL# A2F, ',
     &                ' 10xRho(t),  PL# A3, PL# A3F,' )
         If ( m .eq. -1 ) Write ( *, 4002 )
         If ( m .eq.  0 ) Write ( *, 4003 )
         If ( m .ge.  1 ) Write ( *, 4004 )

         Do istep = 0, 7200   ! Run for two hours ...
            time = istep      ! Timesteps are 1 second long
            Rho0(istep) = ScaleF * sqrt(time) /
```

```
     &                    ( 1.0 + (0.001 * time)**3 )
            fluct = sin( 2.0*PI*time/20.0 )  ! Period of 20 seconds ...
            RhoF(istep) = (1.0 + fluct) * Rho0(istep)

            If ( istep .eq. 0 ) Go To 4000  ! Skip computation 0th step ...
            Do k = 1, 3
               SF = 10.0**(k-2)   ! Renormalizes AEGL 1, 2, and 3 (temp!)
               TLoad(istep,k,1) = TLoad(istep-1,k,1)   ! No fluctuations
     &              + TL_RATE( SF * Rho0(istep), k )   ! 1 sec timesteps
               TLoad(istep,k,2) = TLoad(istep-1,k,2)   ! +  fluctuations
     &              + TL_RATE( SF * RhoF(istep), k )   ! 1 sec timesteps
            End Do

 4000       If ( mod(istep, 50) .eq. 0 )
     &         Write (*, 4001) istep, ( Rho0(istep)*10.0**(k-2),
     &            TLoad(istep,k,1), TLoad(istep,k,2), k = 1,3 )

         End Do  ! Loop over timesteps
      End Do  ! loop over agents
c _____________________________________________________________________.__________
 4001 Format ( I5,',', 1X, F9.4,',', F8.3,',', F8.3,',',
     &                 2X, F9.4,',', F8.3,',', F8.3,',',
     &                 2X, F9.4,',', F8.3,',', F8.3,',' )

      Write ( 6, * ) 'EAGLE_DRIVER: End of exercise = ', icase

      Stop
      End Program EAGLE_DRIVER

c  ***************************************************************.*********
```