



ARL-TR-7456 • SEP 2015



**US Army Research Laboratory
Power-Line UAV Modeling and Simulation
(ARL-PLUMS Ver 2.x)
Software Tool:
User Manual and Technical Report**

by Ross N Adelman and David M Hull

Approved for public release; distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



**US Army Research Laboratory
Power-Line UAV Modeling and Simulation
(ARL-PLUMS Ver 2.x)
Software Tool:
User Manual and Technical Report**

by Ross N Adelman and David M Hull
Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) Sep 2015		2. REPORT TYPE Final		3. DATES COVERED (From - To) 6/2013–8/2014	
4. TITLE AND SUBTITLE US Army Research Laboratory Power-Line UAV Modeling and Simulation (ARL-PLUMS Ver 2.x) Software Tool User Manual and Technical Report				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Ross N Adelman and David M Hull				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-SES-P 2800 Powder Mill Road Adelphi, MD 20783-1137				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-7456	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The US Army Research Laboratory Power-Line Unmanned Aerial Vehicle (UAV) Modeling and Simulation (ARL-PLUMS) Software Tool allows a user to model, compute, and analyze the electric and magnetic fields around power lines. ARL-PLUMS comes with an interactive graphical user interface, which accesses a compute engine to calculate these fields around power lines due to various ground and line geometries and load conditions. ARL-PLUMS makes some simplifying assumptions about the ground and line geometries by using a 2-dimensional (2-D) power-line model. By doing so, less time is needed to solve a model. Even with these assumptions, however, the results are excellent for many applications, providing a “90%” solution in only a couple minutes. The user can experiment with the model parameters to obtain a solid understanding of the underlying physics. Version 2.x enhances many features in earlier versions and adds entirely new ones. This user manual and technical report describes these new features and guides the user through using them.					
15. SUBJECT TERMS power line, electric field, magnetic field, ARL-PLUMS, 2-D model, simulation					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 62	19a. NAME OF RESPONSIBLE PERSON David M Hull
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-3140

Contents

List of Figures	v
List of Tables	vii
1. Introduction	1
2. Power-Line Model	3
2.1 Magnetic-Field Model	5
2.2 Electric-Field Model	6
3. Validation of ARL-PLUMS Using Measured Power-Line Data	7
4. Install ARL-PLUMS	9
4.1 Install the Non-Standalone Version	10
4.2 Install the Standalone Version	10
5. License ARL-PLUMS	11
6 Run ARL-PLUMS	13
7. Create the Power-Line Model	15
7.1 Ground Geometry	15
7.2 Line Geometry	18
7.2.1 Create a “Single”	19
7.2.2 Create a “Horizontal Triple”	20
7.2.3 Create a “Vertical Triple”	21
7.2.4 See the Line Geometry and Calculate Voltages, Capacitances, and Linear Charge Densities	22
7.3 Circuits	24
7.4 Loads	25
7.5 Modeling DC Power Lines	27
8. Construct the UAV Path and Calculate Signatures	28
8.1 UAV Path	28

8.2	Create Signatures	29
8.3	Calculate Signatures	30
9.	Create and Calculate Cutting Planes	32
9.1	Create Cutting Planes	32
9.2	Calculate Cutting Planes	34
10.	Accessing ARL-PLUMS Functionality in MATLAB	36
10.1	Check the ARL-PLUMS License	37
10.2	Open an ARL-PLUMS Model	38
10.3	Validate the Model	39
10.4	Compute Capacitances	39
10.5	Compute Linear Charge Densities	39
10.6	Compute Fields	40
10.7	Compute Basis Fields	40
10.8	Compute Time-Varying Fields	42
10.9	Plot a Signature	42
10.10	Plot a Field in a Cutting Plane	43
11.	Using the ARL-PLUMS API: ARL Wire-Avoidance Simulation Program (ARL-WASP)	44
11.1	Novel Power-Line Detection and Avoidance Methods	44
11.1.1	Detection	44
11.1.2	Avoidance	45
11.2	ARL-WASP	46
12.	Additional Information	50
12.1	Feedback	50
12.2	License Information	50
12.3	Disclaimers	50
12.4	Fair Use	50
13.	References	51
	Distribution List	56

List of Figures

Fig. 1	An example 2-D power-line model created in ARL-PLUMS	4
Fig. 2	The ARL-PLUMS model of the Pepco Doubs Conastone Line	8
Fig. 3	The electric (left) and magnetic (right) fields were computed by ARL-PLUMS and plotted (red and blue curves) on top of the real power-line data presented in (Lambdin 1978). These figures (apart from the red and blue curves that we added) were taken from that paper.	9
Fig. 4	An error message one might encounter if the computer does not have the MCR installed	11
Fig. 5	The splash screen. When unlicensed, it looks like the one of the left; when properly licensed, it looks like the one of the right.	11
Fig. 6	The window that appears where the user can create a license	12
Fig. 7	The window that appears when one creates a license	12
Fig. 8	The window that appears where one can enter a key	13
Fig. 9	The window that appears when one enters a key	13
Fig. 10	ARL-PLUMS when loaded	14
Fig. 11	The menu bar	14
Fig. 12	The “Ground Geometry” panel	15
Fig. 13	Parameters for “Ground Only”	16
Fig. 14	Parameters for “Hemicylindrical Ridge”	16
Fig. 15	Parameters for “Left Wall”	16
Fig. 16	Parameters for “Right Wall”	17
Fig. 17	The windows that appear upon clicking “See Ground Geometry” for the 4 different types of ground geometries	17
Fig. 18	The 3 icons in the toolbar that can be used to change the aspect ratio of the plot and save the plot	18
Fig. 19	The “Line Geometry” panel	18
Fig. 20	The popup menu that appears when one clicks “Create Line(s)”	19
Fig. 21	The arrows one can use to scroll through the lines. The label in between the arrows shows which line(s) are currently being edited. ..	19
Fig. 22	The panel and diagram for a “Single”	19
Fig. 23	When one chooses “Auto” for the color of a “Single”, the color is chosen based on V_{phase} . When V_{rms} is zero, the color will be green, no matter the phase.	20
Fig. 24	The panel and diagram for a “Horizontal Triple”	21
Fig. 25	The panel and diagram for a “Vertical Triple”	22

Fig. 26	The window that appears when one clicks “See Lines”	23
Fig. 27	The window that appears when one clicks “Calculate Voltages”. These are the line-to-ground voltages for the 115-kV 3-phase circuit seen in Fig. 24.	24
Fig. 28	The window that appears when one clicks “Calculate Capacitances”. The values shown have units of pF/m. This is the capacitance matrix for the 115-kV 3-phase circuit seen in Fig. 24.....	24
Fig. 29	The window that appears when one clicks “Calculate Lambdas”. These are the linear charge densities for the 115-kV 3-phase circuit seen in Fig. 24. Note how the linear charge densities on the 3 lines are close to, but not quite, 120° out of phase with each other, despite their voltages being exactly 120° out of phase with each other.	24
Fig. 30	The “Circuits” panel	25
Fig. 31	The “Loads” panel	26
Fig. 32	The window that appears when clicking “Calculate Line Currents”. These currents correspond to 10-A resistive base load and three 50-kW motors with a power factor of 0.8 being attached to the 115-kV 3-phase circuit seen in Fig. 24. In this particular example, the base load is $(115 \text{ kV}) * (10 \text{ A}) * \text{sqrt}(3) = 2 \text{ MW}$, which easily dwarfs the three 50-kW motors.	27
Fig. 33	The “UAV Path” panel	28
Fig. 34	The window that appears when one clicks “See UAV Path”	29
Fig. 35	The “Create Signatures” panel.....	30
Fig. 36	The “Calculate Signatures” panel	31
Fig. 37	The signatures for “ E_y, rms ”, “ E_y, phase ”, and “ $E_y(t)$ ” with no noise added.....	32
Fig. 38	The signature for “ $E_y(t)$ ” with noise added.....	32
Fig. 39	The “Create Cutting Planes” panel	33
Fig. 40	The window that appears when one clicks “See Cutting Planes”.....	34
Fig. 41	The “Calculate Cutting Planes” panel.....	35
Fig. 42	The cutting planes for “ E_y, rms ” and “ E_y, phase ”.....	35
Fig. 43	Twelve frames of an example movie	36
Fig. 44	Examples of using the 2 plotting functions.....	43
Fig. 45	The rms magnitude of the electric (left) and magnetic (right) fields around some example power lines. Notice how they form bull’s-eye patterns around the lines.	45
Fig. 46	The ARL-WASP GUI. This is where one enters the simulation parameters and start the simulation.....	47
Fig. 47	The different wire-avoidance algorithms to choose from, which are the same as those described previously	47

Fig. 48	The ARL-WASP simulation in action. In this case, “Fly Over and Turn Downstream” was chosen as the action to take.	48
Fig. 49	The 3-step simulation loop used in ARL-WASP.....	48

List of Tables

Table 1	The 10 functions available in the ARL-PLUMS API.....	37
---------	--	----

INTENTIONALLY LEFT BLANK.

1. Introduction

The US Army Research Laboratory Power-Line Unmanned Aerial Vehicle (UAV) Modeling and Simulation (ARL-PLUMS) Software Tool allows a user to model, compute, and analyze the quasistatic electric and magnetic fields around alternating current (AC) power lines. ARL-PLUMS comes with an interactive graphical user interface (GUI), which accesses a compute engine to calculate these fields around these lines due to various ground and line geometries and load conditions. ARL-PLUMS allows the user to rapidly define all significant model parameters and compute the electric and magnetic fields along a UAV path or in a cutting plane. In addition, a set of false-color plots can be created to show the time-varying nature of the fields as a movie. ARL-PLUMS also comes with an application programming interface (API) for accessing some of these features from MATLAB without using the GUI.

There are many motivations for modeling and simulating the electric and magnetic fields around power lines. For example, one may want to estimate the magnitudes of these fields near power-line support structures or along the edges of a right of way for different wire configurations. One may also want to build sensors for detecting and localizing power lines in order to facilitate power-line-aided navigation. This requires knowing the behavior and structure of the fields, which ARL-PLUMS can help to provide. Other examples include wire avoidance during UAV operations, power sensing during disaster recovery, and energy harvesting.

ARL-PLUMS makes some simplifying assumptions about the ground and line geometries by using a 2-dimensional (2-D) power-line model. By doing so, less time is needed to define a model and compute the fields. Five modeling assumptions are made:

- 1) The ground can only take the form of a handful of geometries (options are detailed in Section 7).
- 2) The lines are parallel to the x axis. That is, they are perfectly straight and do not sag. They are parallel to the ground and each other.
- 3) There are no power poles or any other conducting objects near the lines.
- 4) The ground is assumed to have a high-enough conductivity so that it “shorts out” the electric field and behaves like an equipotential surface, but a low-enough conductivity so that the secondary magnetic field due to the induced eddy currents in the ground is negligible.

- 5) The ground is assumed to be non-magnetic (i.e., has a relative permeability of one).

A complete description of the 2-D power-line model can be found in Section 2.

Even with these assumptions, however, the results are excellent for many applications, providing a “90%” solution in only a few minutes. The user can experiment with different ground and line geometries and load conditions to obtain a solid understanding of the underlying physics. There are dozens of parameters the user can adjust. Among these are the type of ground geometry; the position and orientation of the lines; the radii of the lines; the line voltages and currents (rms magnitudes and phases); and the operating frequency of the lines.

ARL-PLUMS stores the voltages and currents on the lines, as well as the electric and magnetic fields around the lines, as phasors. A phasor is a root mean square (rms) magnitude and phase, which, along with a frequency, uniquely determines a sinusoidal signal. This allows the user to easily investigate what the electric and magnetic fields look like due to each line separately, and what kind of cancellations occur from all of the lines together. ARL-PLUMS also computes the electric and magnetic fields in the time domain when calculating signatures and generating movies.

Once the model parameters are chosen, the user can compute and visualize the electric and magnetic fields in a number of ways:

- by calculating time-domain signatures for the electric and magnetic fields along a UAV path;
- by calculating the electric and magnetic fields in a cutting plane perpendicular to the wires; and
- by generating a movie that shows the time-varying electric and magnetic fields in a cutting plane over one complete AC power cycle.

ARL-PLUMS 2.x enhances many features in ARL-PLUMS 1.0 (Hull 2012) and earlier versions and adds several entirely new features:

- 1) The user is no longer limited to how many lines can be included in a model.
- 2) The user can specify standard 3-phase circuits (Delta and Wye) in addition to single conductors.
- 3) In addition to specifying the line currents directly, the user can attach loads across conductors, and ARL-PLUMS will compute the resulting line currents for those loads.

- 4) The ground and line geometries, UAV path, and cutting plane dimensions are plotted as the user enters parameters.
- 5) The user can create arbitrary, possibly nonlinear, UAV paths.
- 6) The user can select which signatures and cutting planes to compute and plot as opposed to them all being rendered by default.
- 7) As mentioned previously, ARL-PLUMS 2.x includes an API that allows the user to access many of the features in ARL-PLUMS from a separate modeling and simulation program.

This user manual and technical report describes these new features and guides the user through using them. Section 2 describes the mathematical model that is used in ARL-PLUMS to compute the 2-D electric and magnetic fields in the vicinity of the power lines. Section 3 describes how we validated this model using actual field measurements made by EPA under a 510-kV transmission line. Sections 4 and 5 describe a one-time procedure needed to set up ARL-PLUMS on a new PC. Sections 6–9 describe how to operate ARL-PLUMS at the GUI level. Sections 10–11 describe how to operate ARL-PLUMS at the API level. Sections 12 and 13 provide pointers to additional information.

2. Power-Line Model

ARL-PLUMS uses a right-handed coordinate system. When seated at the computer and looking at the monitor, the positive x direction is out of the screen, toward the user; the positive y direction is to the right; and the positive z direction is up, toward the ceiling. The power lines run parallel to the x axis and positive current runs in the positive x direction. Sea level is always at $z = 0$ m, and all other geometry is referenced to this. An example 2-D power-line model created in ARL-PLUMS is shown in Fig. 1. In this example, there is a single 3-phase circuit 20 m above the ground, and the wires are spaced 4.4 m apart.

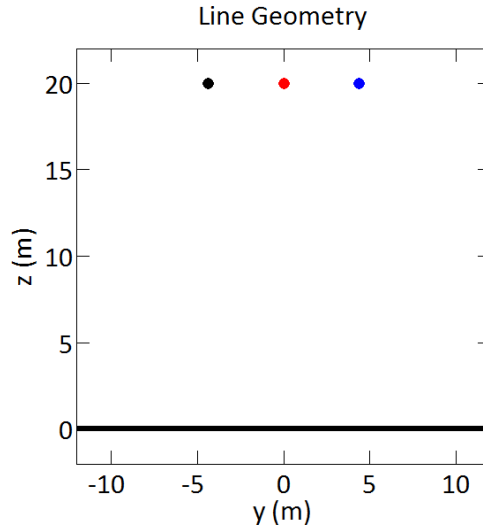


Fig. 1 An example 2-D power-line model created in ARL-PLUMS

Generally, the electric and magnetic fields around power lines are treated separately and independently as quasistatic entities. This is permissible because at typical power-line frequencies (50 or 60 Hz) or low (e.g., 3rd or 5th) harmonics, the corresponding electromagnetic wavelengths are 1,000–6,000 km. We are only interested in the behavior of the fields in the vicinity of the power lines. This is generally 0–10 km away, which is a fraction of these wavelengths. In this region, coupled electromagnetic effects (such as wave propagation) are negligible compared to the effects of the quasistatic sources, themselves. For a deeper discussion on when the quasistatic approximation is applicable in general systems, see Woodson (1968). For a more complete explanation on why the quasistatic approximation can be accurately applied to power systems, such as the ones used in ARL-PLUMS and described in this report, see Olsen (1992).

In the quasistatic case, the voltages on the lines give rise to an electric field around the lines, and the currents on the lines give rise to a magnetic field around the lines. However, while the currents are the sources of the magnetic field, the voltages do not directly source the electric field. Instead, through capacitive coupling, the voltages induce linear charge densities on the lines and an image charge distribution on the ground, and this charge distribution is the actual source of the electric field. In either case, the total (measured) field at any point is the superposition of the constituent fields from each line source at that point.

In the case of the electric field, the closed-form analytical models in the literature (Olsen 1992) do not readily provide the tools to model more than a single line. Therefore, the temptation is to oversimplify a multi-line power line using the single-wire model. However, the results obtained from these simplified models are generally many times larger than the actual fields from the multi-line system being

modeled. For example, the electric field due to a 3-phase power line rotates in space, while the electric field due to a single-phase line pulses (in phase) with the line voltage. Additionally, since the 3 constituent fields are 120° out of phase with each other, the total field is a fraction (typically on the order of 10%) of the constituent fields. The power-line model in ARL-PLUMS solves these issues by modeling the line-to-line and line-to-ground capacitances, which are needed to more accurately model multi-line configurations.

2.1 Magnetic-Field Model

The currents on the power lines give rise to a magnetic field. The horizontal and vertical components of the magnetic field are given by the Biot-Savart Law [Jackson, 1999]:

$$H_y = -\frac{1}{2\pi} \sum_n I_n \frac{z-z_n}{(y-y_n)^2+(z-z_n)^2} \quad (1)$$

$$H_z = \frac{1}{2\pi} \sum_n I_n \frac{y-y_n}{(y-y_n)^2+(z-z_n)^2} \quad (2)$$

where I_n is the current on the n th line. In other words, the total horizontal and vertical magnetic field components are the superposition of the constituent magnetic field components due to the currents on each line. Since the fields are quasistatic, the magnetic fields given by Eqs. 1 and 2 can be computed at any instant of time, using a set of instantaneous line currents I_n . These fields can also be computed as functions of time, if the source currents are expressed as functions of time, or as phasors, if the source currents are expressed as phasors.

Note that the phase angle in each phasor is relative to a time reference; it is not related to the geometric angle of the field. Each line n gives rise to a constituent field that is in phase with the alternating current on that line. However, these constituent fields are out of phase with each other in time, so the total field will rotate in time.

Because the 2-D power-line model is symmetric along the x axis, the component of the magnetic field parallel to the lines (along the x axis) is zero. In 3-D models, H_x is typically nonzero (though small compared to H_y and H_z) due to wire sag, branch wires, and other 3-D effects.

These equations are valid as long as the skin depth of the ground is large compared to the height of the wires. The skin depth, δ , is given by (Jackson 1993)

$$\delta = \sqrt{\frac{\rho}{\pi\mu f}} \quad (3)$$

where ρ is the resistivity of the ground, μ is the absolute permeability of the ground, and f is the frequency. For ground resistivities greater than 10 $\Omega\cdot\text{m}$, and at power-line frequencies, the skin depth is greater than 200 m. Since most ground materials have resistivities between 10 and 1000 $\Omega\cdot\text{m}$, the skin depth is greater than 10 times the height of the power lines. Thus, the effects of ground conductivity can be neglected in the magnetic-field model.

2.2 Electric-Field Model

The ground is conducting enough (typically 8–12 orders of magnitude more conductive than the air above it) to support the movement of image charges (that is, image currents) with negligible voltage drops. Thus, the ground “shorts out” the electric field. For this reason, the Earth is modeled as an infinite, perfectly conducting ground plane. The lines are modeled as infinitely long, perfectly conducting wires parallel to the ground and each other.

Power poles and other nearby conducting objects are not included in this model. In general, these objects are conducting enough that they will also short out the electric field. In this case, it would make sense to model them as perfect electric conductors (PEC) at ground potential, and this is normally done in detailed 3-dimensional (3-D) models. However, this cannot be done within a 2-D modeling framework, and so the resulting distortions of the power-line electric field near the poles are not included in the results computed by ARL-PLUMS 2.x.

Using the method of images (Jackson 1993), the horizontal (along the y axis) and vertical (along the z axis) components of the electric field are computed using

$$E_y = \frac{1}{2\pi\epsilon_0} \sum_n \lambda_n \left(\frac{y-y_n}{(y-y_n)^2+(z-z_n)^2} - \frac{y-y_n}{(y-y_n)^2+(z+z_n)^2} \right) \quad (4)$$

$$E_z = \frac{1}{2\pi\epsilon_0} \sum_n \lambda_n \left(\frac{z-z_n}{(y-y_n)^2+(z-z_n)^2} - \frac{z+z_n}{(y-y_n)^2+(z+z_n)^2} \right) \quad (5)$$

where ϵ_0 is the permittivity of free space; y_n and z_n are the horizontal offset and the height of the n th line; λ_n is the linear charge density on the n th line; and y and z are the horizontal offset and the height of the evaluation point.

Note that these equations each have 2 terms: the first term is the field due to the line charge, and the second term is the field due to the image charge. The method of images (Jackson 1999) allows us to replace the PEC ground with an image charge; this enables us to compute the electric field in closed form. The method of images can be extended to other boundary conditions (Jeans 1951); these are discussed in Section 7.1.

In these equations, the ground is assumed to be at $z = 0$ m, but in ARL-PLUMS, the ground can be offset from sea level, which is always at $z = 0$ m. Because of symmetry, the component of the electric field parallel to the lines (along the x axis) is zero.

For the single-wire case, the linear charge density, λ , for a given voltage, V , is given by (Olsen 1992)

$$\lambda = \frac{2\pi\epsilon_0 V}{\ln\left(\frac{2z}{a}\right)} \quad (6)$$

where z is the height of the wire and a is the radius of the wire. For the multi-wire case, this expression needs to be extended to multiple wires by taking into account the line-to-line and line-to-ground capacitances. According to the method of moments (Harrington 1967, Hull 1997), the voltage on each line is the superposition of the voltages due to the linear charge densities on each of the lines. In other words,

$$V_m = \sum_n A_{mn} \lambda_n \quad (7)$$

where

$$A_{mn} = -\frac{1}{4\pi\epsilon_0} \ln \left(\frac{(y_m - y_n)^2 + ((z_m - a_m) - z_n)^2}{(y_m - y_n)^2 + ((z_m - a_m) + z_n)^2} \right) \quad (8)$$

and a_n is the radius of the n th wire and there are N lines. These equations form a system of N linear equations in N unknowns, which can be inverted in a fraction of a second using a laptop personal computer (PC).

Like magnetic fields, the electric-field components can be computed separately and then summed vectorially. Additionally, the constituent fields (due to each line) can be computed, either as functions of time or as phasors, and then summed to provide the total field.

3. Validation of ARL-PLUMS Using Measured Power-Line Data

We validated the 2-D power-line model in ARL-PLUMS using real power-line magnetic and electric-field measurements collected by the Environmental Protection Agency (EPA) under the Pepco Doubs Conastone Line near Frederick, Maryland (Lambdin 1978). The line was a 510-kV, 3-phase circuit. The individual lines were strung in a horizontal configuration, 14.1 m above the ground and 10.7 m apart. Each phase was carried by a pair of conductors, separated by 45.7 cm, and the conductors all had a diameter equal to 4.6 cm.

The EPA researchers reported there was 567 A in each phase. They measured the vertical component of the electric field and the total magnetic flux density along a path perpendicular to the lines. They started 200 ft (61 m) away from the lines, passed under the lines, and continued for 20 ft (6.1 m) beyond the centerline, toward the other side of the lines. They measured these fields along the same path several times, varying how high above the ground they held the sensors. They measured electric and magnetic fields at 2 heights: 1.0 and 1.8 m.

We assumed that the currents in the 3 lines were balanced and that they were in phase with the voltages; that the rms currents did not change with time, and that the heights of the wires did not change with time. We further assumed that the physical ground and conducting ground were at the same altitude, and that the ground had no significant ferroelectric or ferromagnetic properties and had a relative permeability of one. With these assumptions in mind, we modeled the Pepco Doubs Conastone Line in ARL-PLUMS (Fig. 2). ARL-PLUMS normally draws the lines so that they can be easily seen on a PC monitor; the actual wire diameter is usually smaller than the diameter shown.

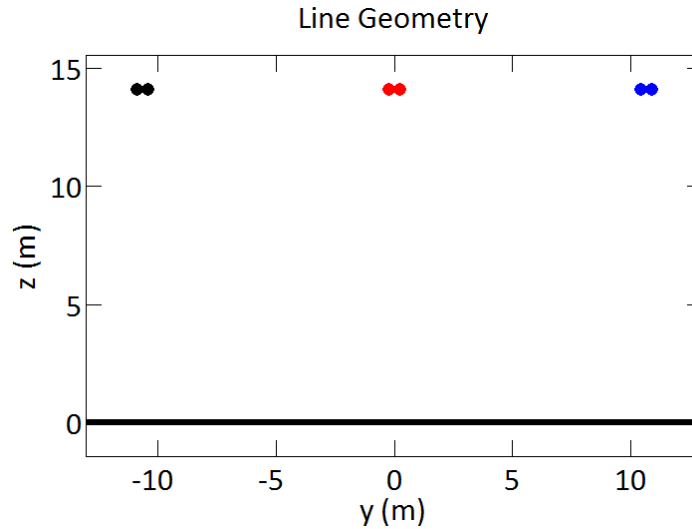


Fig. 2 The ARL-PLUMS model of the Pepco Doubs Conastone Line

Using ARL-PLUMS, we computed the electric field and the total magnetic flux density for this model using the same procedure that they used to measure the fields (we computed the fields at 1.0 m above the ground). We plotted the computed fields on top of the plots that the EPA researchers included in their report; our simulated fields are the red curves seen in Fig. 3. In the left plot, which shows the vertical component of the electric field, our computed values match up very well to their measured values. In the right plot, which shows the total magnetic flux density, our computed fields followed the same shape as their measured values, but were roughly 10% low. This difference is attributed to one or both of the following

reasons. First, the lines may have sagged after the time the heights were measured, and before the time the fields were measured. This would mean the lines were closer to the sensors than we modeled, which would also lead to a larger magnetic flux density at the sensor locations. Second, the line currents may not have been balanced. If the reported 567 A was the actual current only for one phase, and the others were imbalanced, this could lead to a larger magnetic flux density. For example, we increased the current in the left and right lines by 10%, recomputed the total magnetic flux density, and plotted it (the blue curve seen in the right plot in Fig. 3). In this scenario, the resulting total magnetic flux density was closer to the measured values.

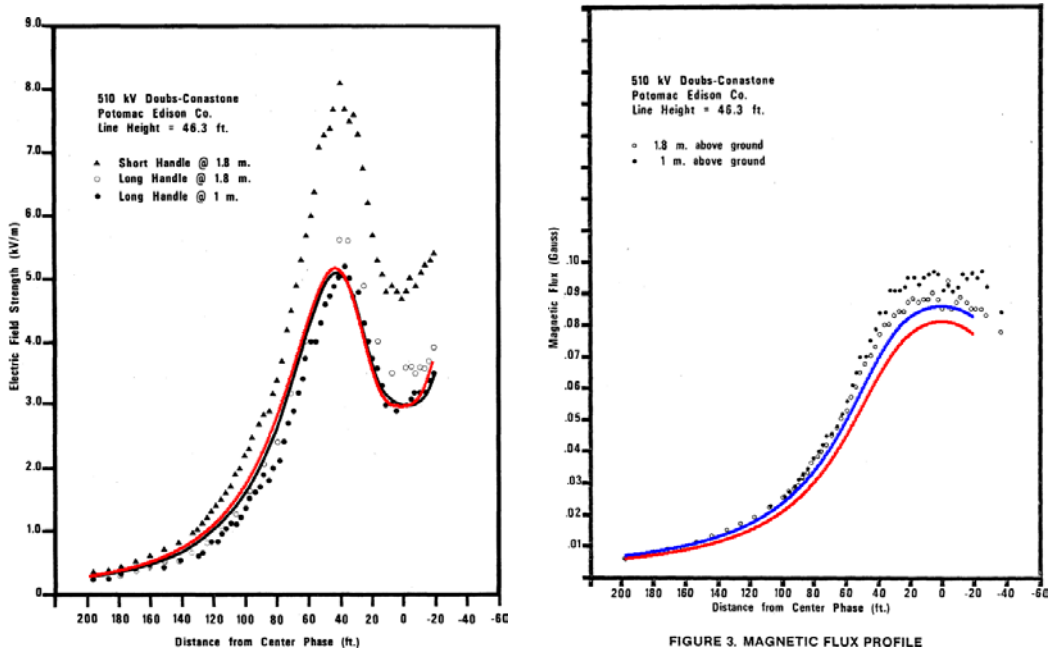


Fig. 3 The electric (left) and magnetic (right) fields were computed by ARL-PLUMS and plotted (red and blue curves) on top of the real power-line data presented in (Lambdin 1978). These figures (apart from the red and blue curves that we added) were taken from that paper.

4. Install ARL-PLUMS

Sections 4 and 5 describe the one-time procedure needed to set up ARL-PLUMS on a new PC.

To install and run ARL-PLUMS, copy `arl_plums2.2.exe` from the ARL-PLUMS CD to a directory on the machine being used. Navigate to that directory and run `arl_plums2.2.exe`. On the window that appears asking the user to unzip the contents, click the button labeled “Unzip”. Upon doing so, the utility will unpack ARL-PLUMS and place it in a new directory called `arl_plums2.2`. Enter that directory.

ARL-PLUMS was developed using MATLAB, which is a numerical computing environment developed by MathWorks (2014). There are 2 methods for running ARL-PLUMS, and depending on the method chosen, ARL-PLUMS requires that either MATLAB or the MATLAB Compiler Runtime (MCR) be installed on the machine.

4.1 Install the Non-Standalone Version

One method, known as the non-standalone version, involves running ARL-PLUMS from inside MATLAB. This method is the easier of the two, especially if MATLAB is already installed on the machine. Additionally, using this method, one can access the ARL-PLUMS API from custom MATLAB code. However, this method will not work unless MATLAB is installed on the machine. To use this method, open MATLAB, and navigate to the directory in which ARL-PLUMS was installed. Run `arl_plums.p` by typing `arl_plums` on the command line and hitting enter on the keyboard. At this point, the splash screen will appear, as detailed in Section 5.

4.2 Install the Standalone Version

The other method does not require MATLAB to be installed on the machine. This method, known as the standalone version, runs separately from MATLAB, but requires the user to install the MCR. The MCR is free and, for convenience, is included on the ARL-PLUMS CD. An advantage of this method is that ARL-PLUMS can run on machines without MATLAB. A downside is that the user must install the MCR. To use this method, navigate to the directory in which ARL-PLUMS was installed, and run `arl_plums.exe`. If an error message like the one shown in Fig. 4 about the machine missing a DLL appears, then the MCR will need to be installed. To install the MCR, launch `MCRInstaller.exe` and follow the instructions. The MCR only takes a couple minutes to install, but requires administrative privileges. This may require calling the IT support desk. When the MCR has been installed, run `arl_plums.exe` again. Note that, because the MCR is large (around 100 MB), `arl_plums.exe` may take a couple of minutes to start up, especially on older or memory-limited PCs. Startup is usually faster on subsequent executions (typically a couple seconds) because the MCR is already loaded into memory. Once ARL-PLUMS is running, the splash screen will appear, as detailed in Section 5.

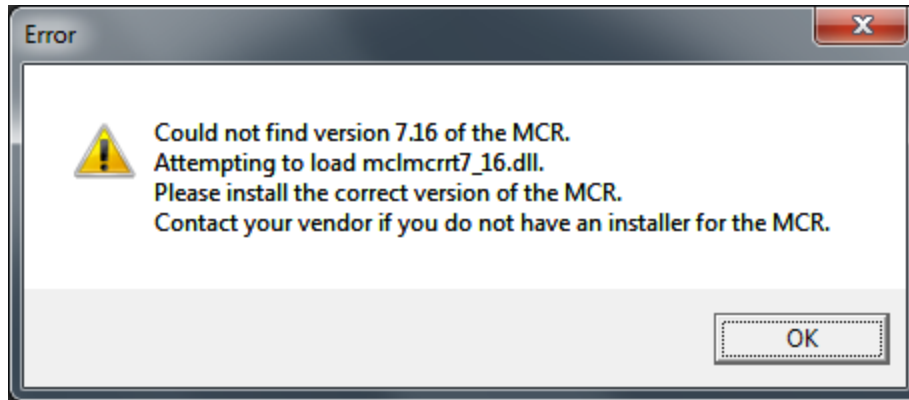


Fig. 4 An error message one might encounter if the computer does not have the MCR installed

5. License ARL-PLUMS

Licenses to use ARL-PLUMS can be provided at no cost to US Government collaborators and their contractors. Generally, a Memorandum of Agreement (MOA), Software License Agreement (SLA), Cooperative R&D Agreement (CRADA), or similar contractual relationship must be in place before licenses can be provided.

When ARL-PLUMS loads, a splash screen (Fig. 5) appears. When unlicensed, the splash screen will include some general license information. To accept them, click “OK”. To decline them, click “Cancel”.

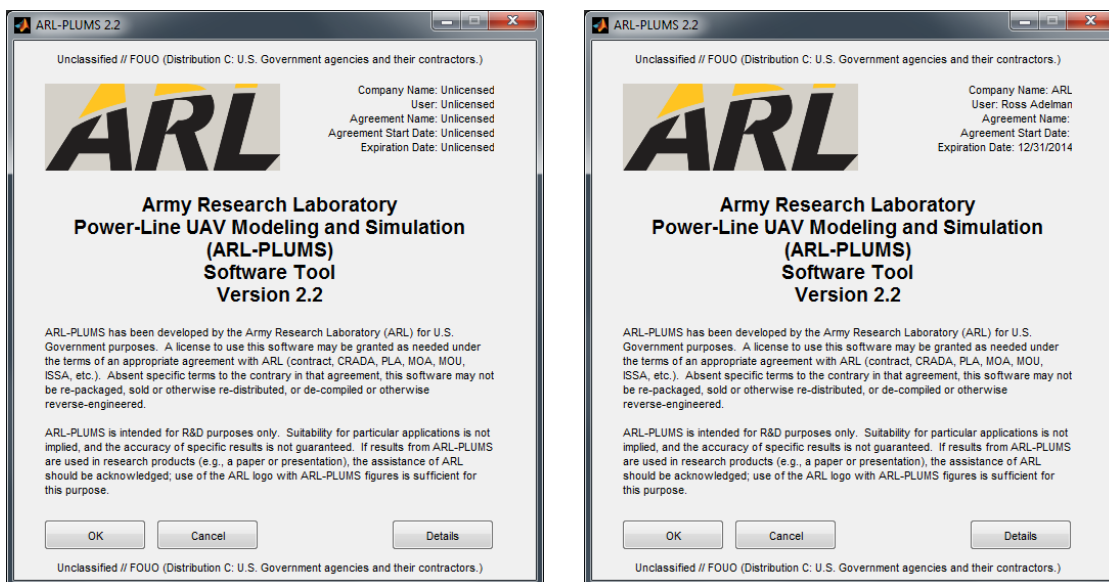


Fig. 5 The splash screen. When unlicensed, it looks like the one of the left; when properly licensed, it looks like the one of the right.

Upon clicking “OK”, a window will appear asking the user to create a license (Fig. 6). The username and computer’s media access control (MAC) address(es) should already be loaded. To complete the form, enter the company name, full name, and the date on which the license should expire. The other boxes are optional. When done, click “Create”.

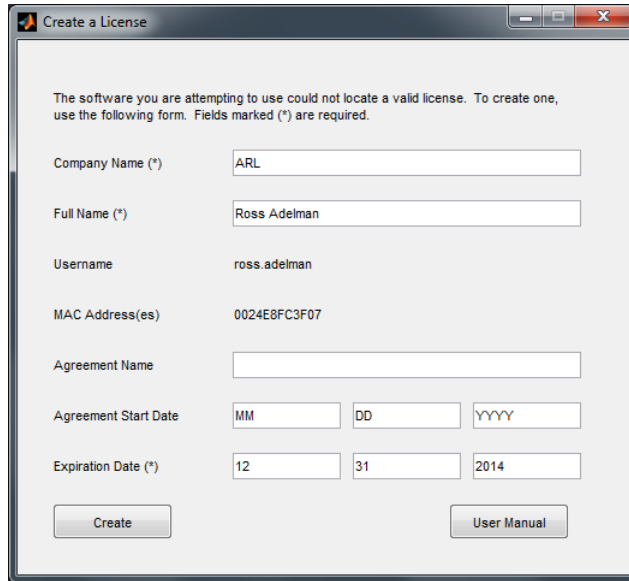


Fig. 6 The window that appears where the user can create a license

A message box will appear telling the user that a license, `licence2.2.txt`, has been created (Fig. 7). Send `licence2.2.txt`, which is located in the `license` directory, or the information contained in it to ARL. ARL will use this information to create a matching key, which can be used to activate ARL-PLUMS.

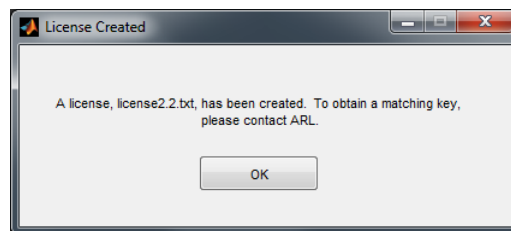


Fig. 7 The window that appears when one creates a license

Once the user has the key, launch ARL-PLUMS again, click “OK” on the splash screen, and enter the key in the window that appears (Fig. 8).

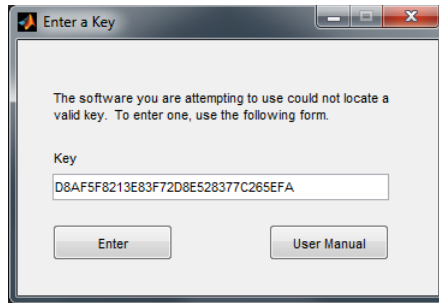


Fig. 8 The window that appears where one can enter a key

Click “OK” when done, and a message box will appear (Fig. 9) telling the user that a key has been entered. Launch ARL-PLUMS again, and, assuming the key was valid, the splash screen will show the license details. Click “OK” to continue on to ARL-PLUMS. The user should only have to go through the procedure shown in Figs. 6–9 once, but the procedure will need to be repeated for additional users, additional PCs, when the license agreement is updated, or when upgrading to a different release of ARL-PLUMS.

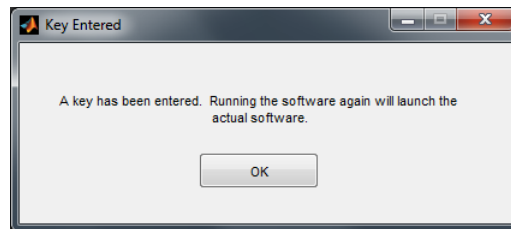


Fig. 9 The window that appears when one enters a key

6 Run ARL-PLUMS

Sections 6–9 describe how to operate ARL-PLUMS at the GUI level. These sections form a basic user manual for those who do not want to program, but do want to model power lines and get usable results, including presentation-quality graphics, in just a few minutes.

When ARL-PLUMS loads (Fig. 10), click “File”, then “Open” (Fig. 11). In the dialog box that appears, navigate to the `models` directory, select `model.mat`, and click “Open”. This will load an example model. To create a new model, click “File”, then “New”. To save a model, click “File”, then “Save As”.

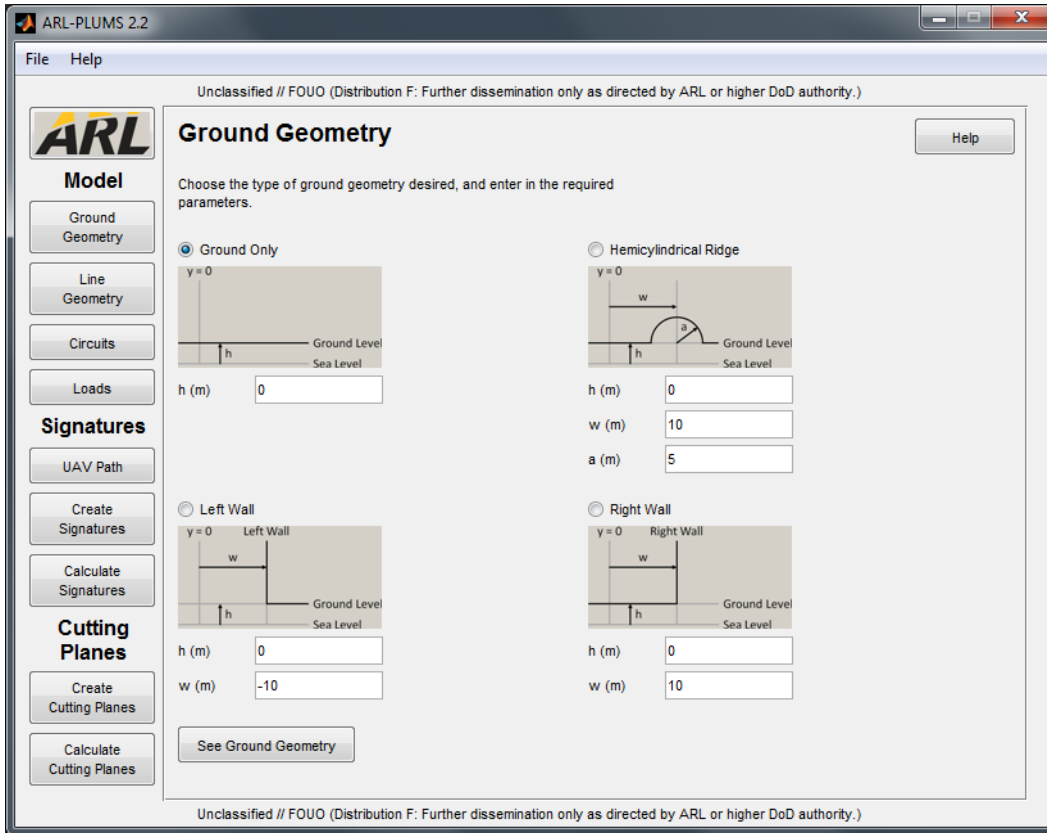


Fig. 10 ARL-PLUMS when loaded

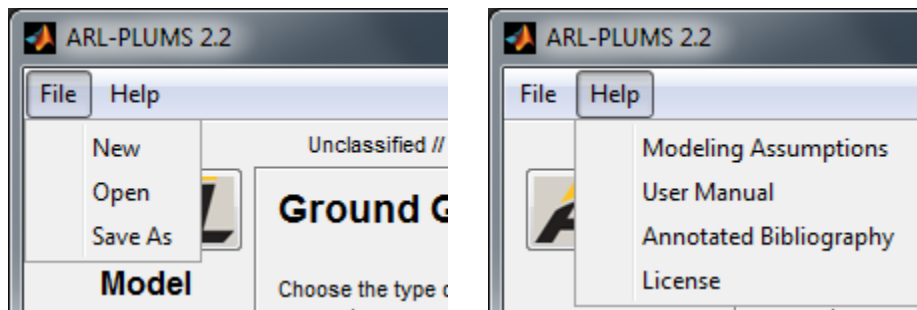


Fig. 11 The menu bar

Help is available. To see a document describing the modeling assumptions made by ARL-PLUMS, click “Help”, then “Modeling Assumptions”. To bring up a copy of the user manual, click “Help”, then “User Manual” (or click “Help” at the top-right corner of any panel). For more information on ARL’s research on power lines, click “Help”, then “Annotated Bibliography”. To see the splash screen again, click “Help”, then “License”.

7. Create the Power-Line Model

7.1 Ground Geometry

The “Ground Geometry” panel (Fig. 12) is where the user constructs the ground beneath the power lines.

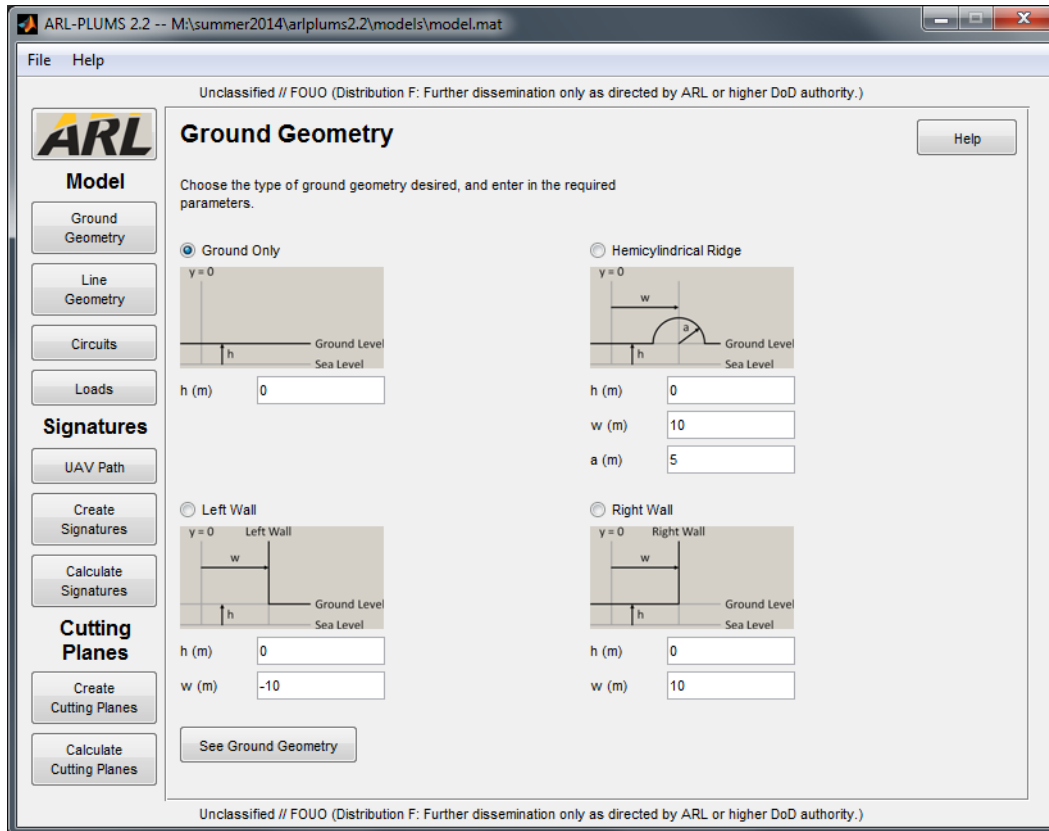


Fig. 12 The “Ground Geometry” panel

There are 4 types of ground geometries to choose from:

- “Ground Only”: the ground is perfectly flat.
- “Hemicylindrical Ridge”: the ground is perfectly flat with a single hemicylindrical ridge that runs parallel to the power lines. This could be useful for modeling a tree line next to the power lines. Only one hemicylindrical ridge can be modeled at any one time (i.e., 2 or more hemicylindrical ridges cannot be modeled simultaneously).
- “Left Wall”: the ground is perfectly flat with a perfectly flat wall to the left. The ground and the wall come together at a right angle, and the problem

domain opens up and to the right. This could be useful for modeling a tall building next to the power lines.

- “Right Wall”: the ground is perfectly flat with a perfectly flat wall to the right. The ground and wall come together at a right angle, and the problem domain opens up and to the left.

The ground, hemicylindrical ridge, and walls are assumed to be perfectly conducting in the electric-field model. Because the induced currents are so small, this is a reasonable assumption for practically all building materials. These features are not used in the magnetic-field model; that is, the ground geometry is assumed to have no effect on the magnetic field. Unless the ground is loaded with ferromagnetic materials or a building has a large steel structure, this is also a reasonable assumption. Select the radio button by the type of ground geometry wanted and enter the parameters as shown in the diagrams. Figs. 13–16 display the parameters for each type of ground geometry.



Fig. 13 Parameters for “Ground Only”

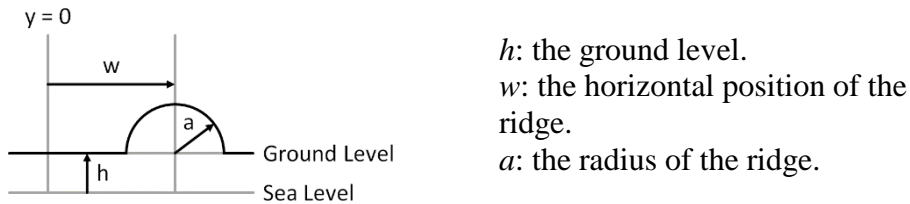


Fig. 14 Parameters for “Hemicylindrical Ridge”

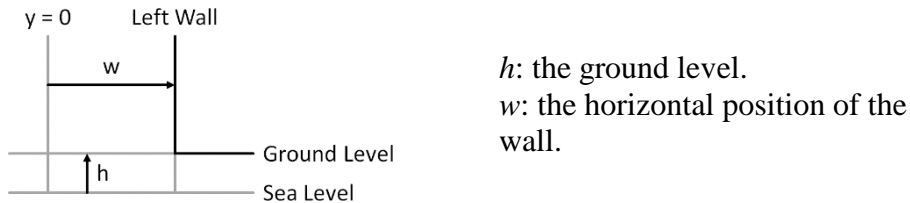
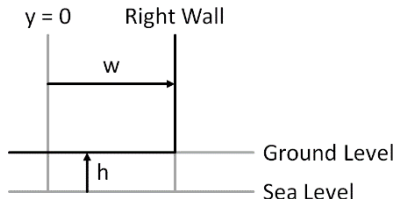


Fig. 15 Parameters for “Left Wall”



h : the ground level.
 w : the horizontal position of the wall.

Fig. 16 Parameters for “Right Wall”

Ground level for each of the 4 types of ground geometries is referenced to sea level, which is always at $z = 0$ m. After entering the parameters, click “See Ground Geometry”, and a window will appear showing what the ground geometry looks like (Fig. 17). The window updates as the user modifies the parameters in the panel.

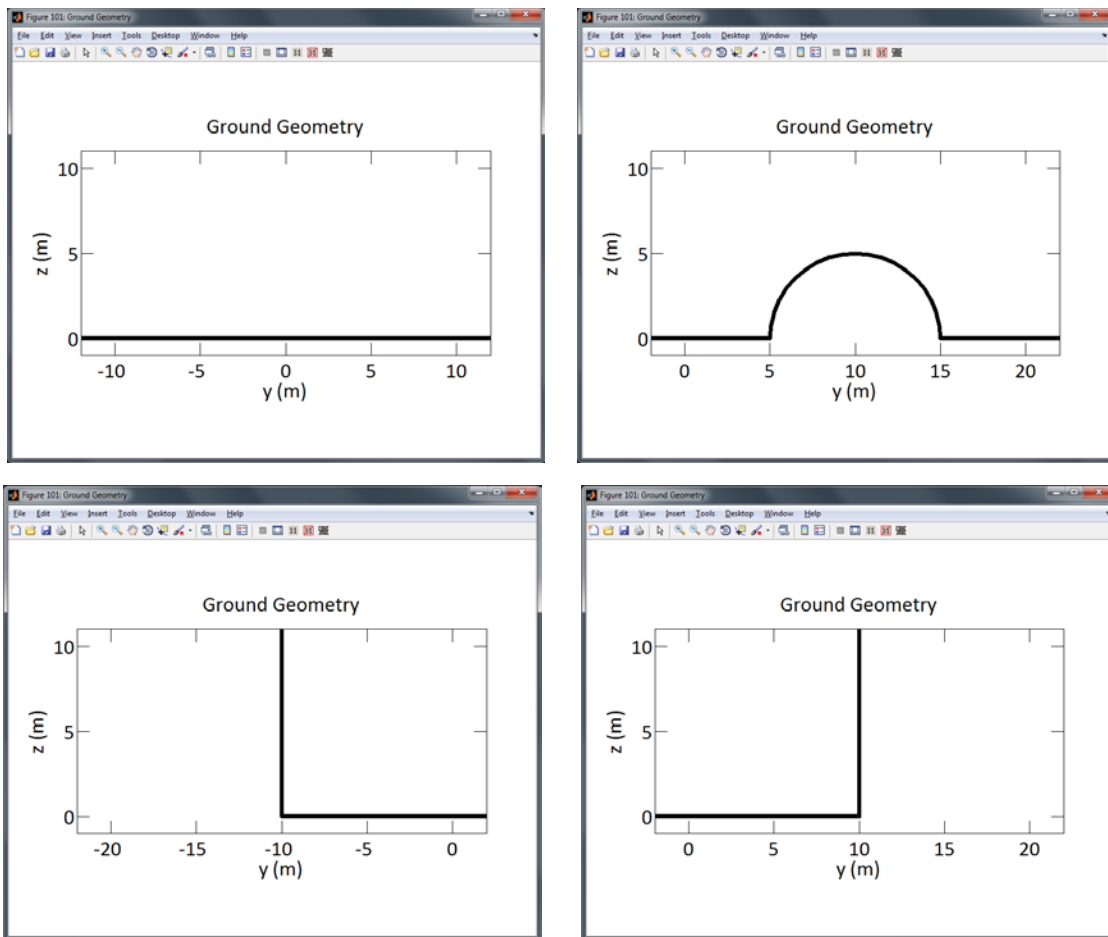


Fig. 17 The windows that appear upon clicking “See Ground Geometry” for the 4 different types of ground geometries

There are 3 icons in the toolbar on the window that displays the ground geometry (Fig. 18). Click to use a 1-to-1 aspect ratio when plotting; click to allow the plot to expand to size of the window (i.e., do not use a 1-to-1 aspect ratio); and click

SAVE to save the plot. Most plots that ARL-PLUMS generates have some or all of these icons available for use.



Fig. 18 The 3 icons in the toolbar that can be used to change the aspect ratio of the plot and save the plot

7.2 Line Geometry

The “Line Geometry” panel (Fig. 19) is where one constructs the power-line model. This panel is also where one specifies the operating frequency of the power lines. This is typically 50 Hz, 60 Hz, or one of their harmonics. Note that ARL-PLUMS is capable of modeling direct current (DC) power lines. To do this, enter “0” for the operating frequency. However, ARL-PLUMS was designed with the AC case in mind, so doing this can lead to quirky behavior and possibly misleading results. Be cautious when doing so. See Section 7.5 for more information.

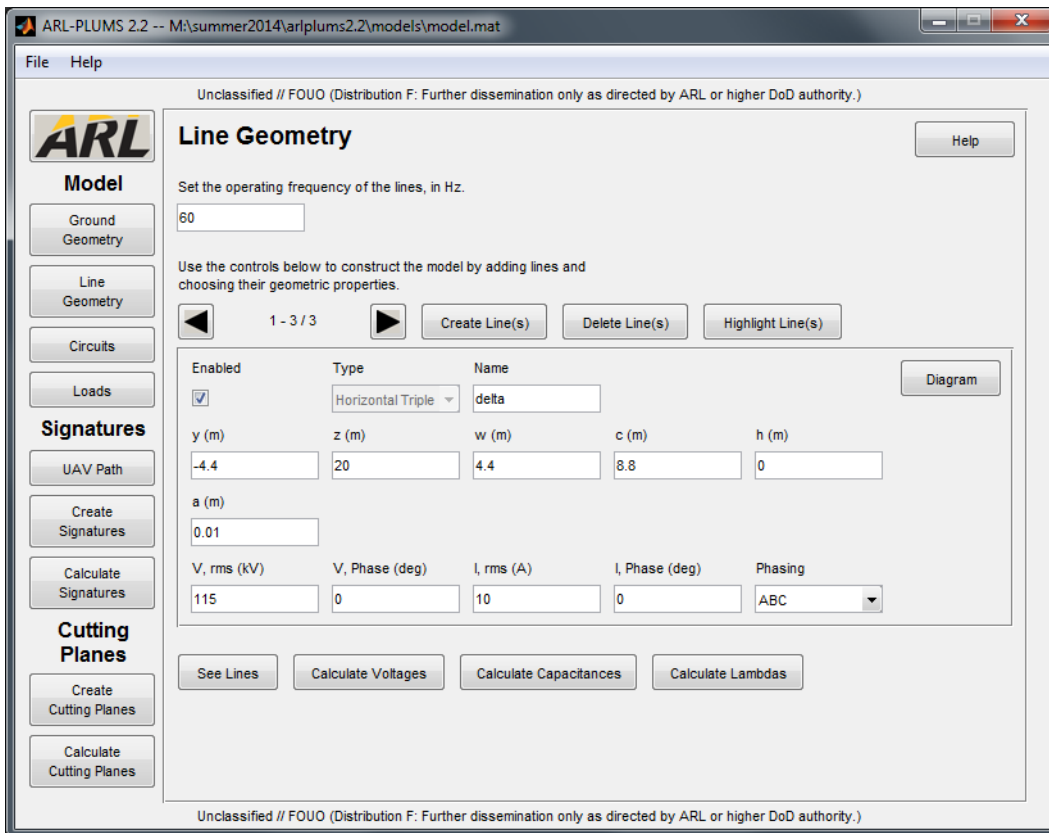


Fig. 19 The “Line Geometry” panel

After choosing an operating frequency, the user can begin adding power lines. One can build the model line by line using single conductors or add balanced 3-phase

circuits. To do this, click “Create Line(s)”, and a popup menu will appear (Fig. 20). Click on one of the 3 options to add a line (or lines). After adding lines, one can scroll through them using the arrows (Fig. 21). In between the arrows, there is a label showing which line(s) is currently being edited.

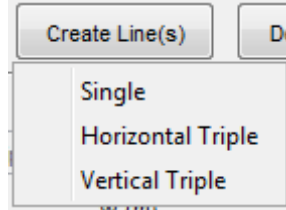


Fig. 20 The popup menu that appears when one clicks “Create Line(s)”



Fig. 21 The arrows one can use to scroll through the lines. The label in between the arrows shows which line(s) are currently being edited.

7.2.1 Create a “Single”

Click “Single” to append a single conductor to the model. Navigate to the newly created panel (Fig. 22), where one can type in the parameters to control the position, radius, voltage, and base current on the “Single”. Click “Diagram” to bring up a window with a diagram showing what each of the parameters mean (Fig. 22).

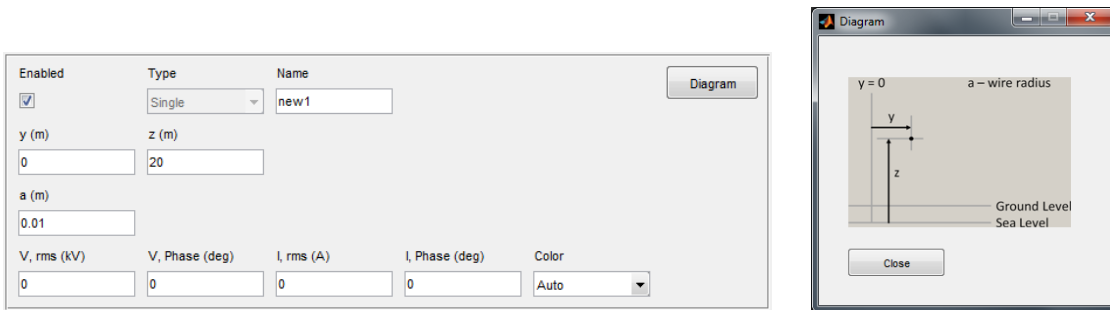


Fig. 22 The panel and diagram for a “Single”

The parameters for a “Single” are the following:

- Enabled: whether the “Single” is included in the model when calculating the electric and magnetic fields.
- Name: the name of the “Single”.
- y : the horizontal position, in m, of the line.
- z : the altitude, in m, of the line above sea level.

- a : the radius, in m, of the line.
- V_{rms} : the rms magnitude, in kV, of the line-to-ground voltage.
- V_{phase} : the phase, in degrees, of the line-to-ground voltage relative to the global phase reference.
- I_{rms} : the rms magnitude, in A, of the base current on the line.
- I_{phase} : the phase, in degrees, of the base current on the line relative to the global phase reference.
- Color: the color on the line (choosing “Auto” allows ARL-PLUMS to automatically pick a color based on V_{phase} as shown in Fig. 23).

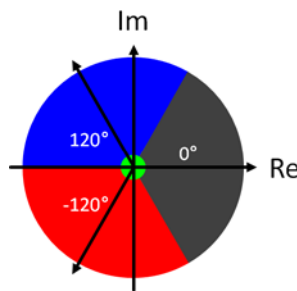


Fig. 23 When one chooses “Auto” for the color of a “Single”, the color is chosen based on V_{phase} . When V_{rms} is zero, the color will be green, no matter the phase.

7.2.2 Create a “Horizontal Triple”

Click on “Horizontal Triple” to append a 3-phase circuit in a horizontal configuration to the model (Fig. 24).

The parameters for a “Horizontal Triple” are the following:

- Enabled: whether the “Horizontal Triple” is included in the model when calculating the electric and magnetic fields.
- Name: the name of the “Horizontal Triple”.
- y : the horizontal position, in m, of the top-left line.
- z : the altitude, in m, of the top-left line above sea level.
- w : the horizontal position, in m, of the top-right line, using the top-left line as a reference.
- c : the horizontal position, in m, of the bottom line, using the top-left line as a reference.
- h : the distance, in m, of the bottom line below the top-left line.

- a : the radius, in m, of the lines.
- V_{rms} : the rms magnitude, in kV, of the line-to-line voltage.
- V_{phase} : the phase, in degrees, of the line-to-ground voltage on line A relative to the global phase reference.
- I_{rms} : the rms magnitude, in A, of the base current on all 3 lines.
- I_{phase} : the phase, in degrees, of the base current on line A relative to the global phase reference. The current on lines B and C are phase-shifted versions of the current on line A.
- Phasing: the relative phasing of the lines (e.g., “ABC” is: the top-left line is line A, the top-right line is line B, and the bottom line is line C).

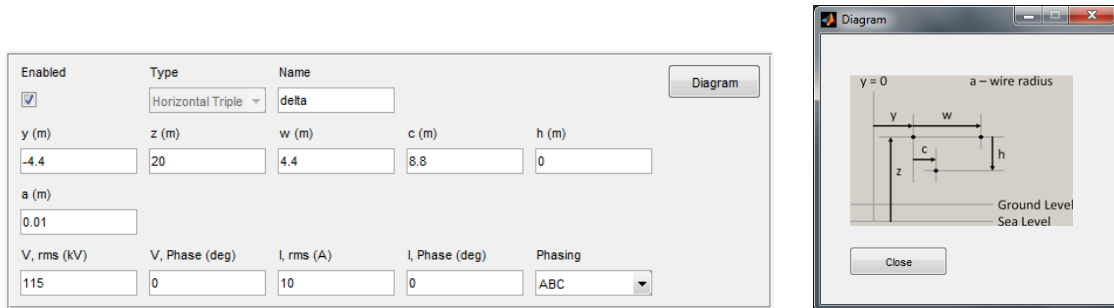


Fig. 24 The panel and diagram for a “Horizontal Triple”

7.2.3 Create a “Vertical Triple”

Click on “Vertical Triple” to append a 3-phase circuit in a vertical configuration to the model (Fig. 25).

The parameters for a “Vertical Triple” are the following:

- Enabled: whether the “Vertical Triple” is included in the model when calculating the electric and magnetic fields.
- Name: the name of the “Vertical Triple”.
- y : the horizontal position, in m, of the top-left line.
- z : the altitude, in m, of the top-left line above sea level.
- w : the distance, in m, of the bottom-left line below the top-left line.
- c : the distance, in m, of the right line below the top-left line.
- h : the horizontal position, in m, of the right line, using the top-left line as a reference.

- a : the radius, in m, of the lines.
- V_{rms} : the rms magnitude, in kV, of the line-to-line voltage.
- V_{phase} : the phase, in degrees, of the line-to-ground voltage on line A relative to the global phase reference.
- I_{rms} : the rms magnitude, in A, of the base current on all 3 lines.
- I_{phase} : the phase, in degrees, of the base current on line A relative to the global phase reference. The current on lines B and C are phase-shifted versions of the current on line A.
- Phasing: the relative phasing of the lines (e.g., “ABC” is: the top-left line is line A, the bottom-left line is line B, and the right line is line C).

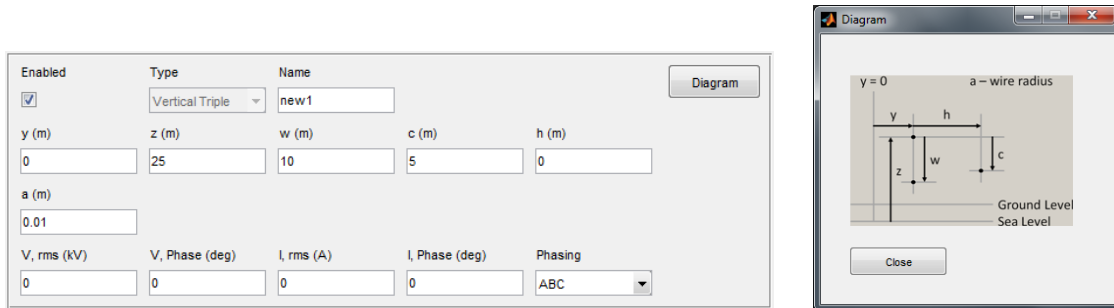


Fig. 25 The panel and diagram for a “Vertical Triple”

Three-phase circuits, such as a “Horizontal Triple” or “Vertical Triple”, are convenient ways to manage 3 lines at once. When looking at one, the label in Fig. 21 shows multiple lines, e.g., “1 - 3 / 3”. The names of the 3 lines are equal to the name of the 3-phase circuit plus “A”, “B”, and “C”. For example, a “Horizontal Triple” called “delta”, like in Fig. 24, would have 3 lines named “deltaA”, “deltaB”, and “deltaC”.

7.2.4 See the Line Geometry and Calculate Voltages, Capacitances, and Linear Charge Densities

To see the line geometry, click “See Lines”. A window will appear showing the line geometry along with the ground geometry (Fig. 26). The window updates as one changes the line geometry. Enabled lines are dark, colored-in circles, whereas disabled lines are hollow, empty circles. To assist in modeling, one can click “Highlight Line(s)” to highlight the line(s) that are currently being edited.

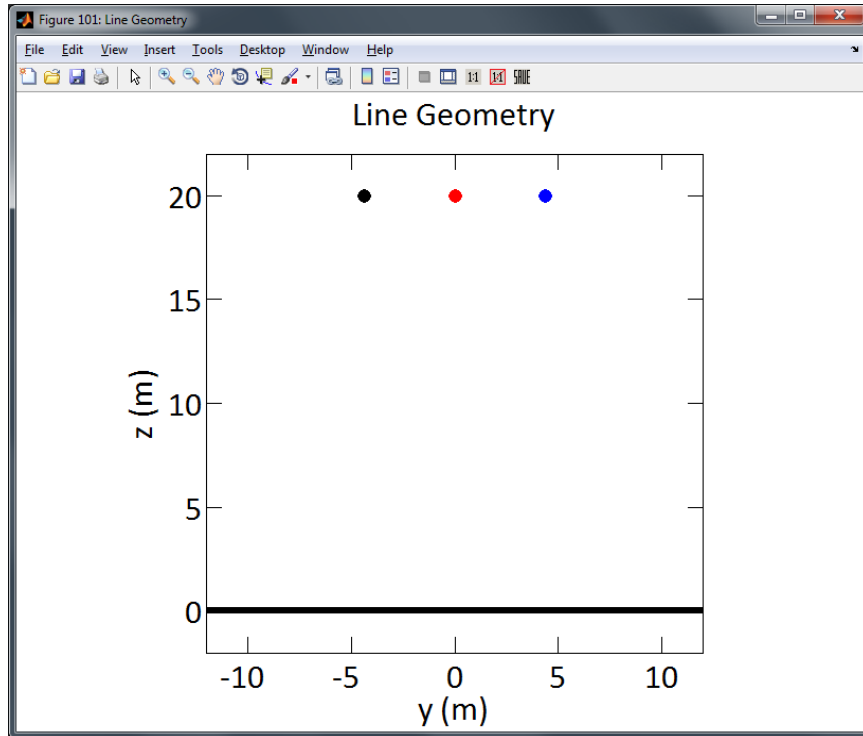


Fig. 26 The window that appears when one clicks “See Lines”

Click “Calculate Voltages” to calculate the line-to-ground voltages of each conductor in the model (Fig. 27). Click “Calculate Capacitances” to calculate the capacitance matrix (Fig. 28). The diagonal entries are called the coefficients of capacitance, and the non-diagonal entries are called the coefficients of induction (Cheng 1992, Jackson 1993). Click “Calculate Lambdas” to calculate the linear charge densities on the lines (Fig. 29). These are computed using

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} = C \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} \quad (9)$$

where λ_k and V_k are the linear charge density and line-to-ground voltage of line k , respectively, and C is the capacitance matrix from before.

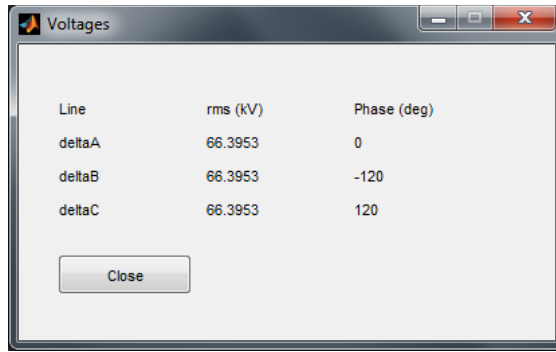


Fig. 27 The window that appears when one clicks “Calculate Voltages”. These are the line-to-ground voltages for the 115-kV 3-phase circuit seen in Fig. 24.

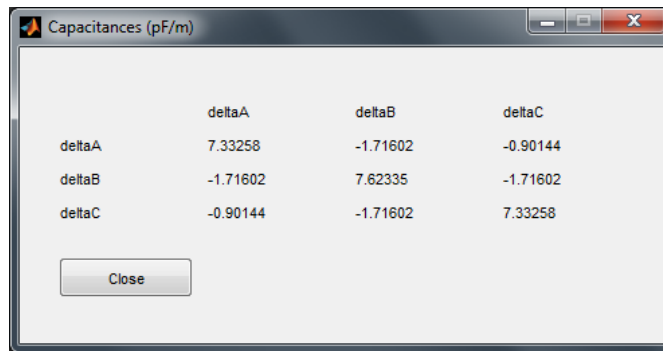


Fig. 28 The window that appears when one clicks “Calculate Capacitances”. The values shown have units of pF/m. This is the capacitance matrix for the 115-kV 3-phase circuit seen in Fig. 24.

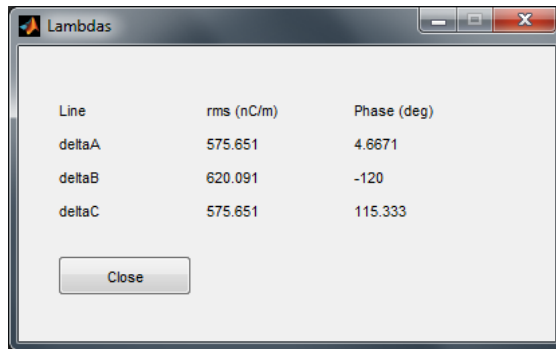


Fig. 29 The window that appears when one clicks “Calculate Lambdas”. These are the linear charge densities for the 115-kV 3-phase circuit seen in Fig. 24. Note how the linear charge densities on the 3 lines are close to, but not quite, 120° out of phase with each other, despite their voltages being exactly 120° out of phase with each other.

7.3 Circuits

The “Circuits” panel (Fig. 30) is where one links conductors together into circuits, to which one can then attach loads. Use the up and down arrows to scroll through the circuits. To delete the circuit in the middle row, click “Delete”. To add a circuit

above or below the circuit in the middle row, click “Add Above” or “Add Below”, respectively. Give each circuit a name, and choose from the drop-down boxes which lines to use as the positive and negative terminals. For example, in Fig. 30, “crcA” is created by linking together “deltaA” and “deltaB”.

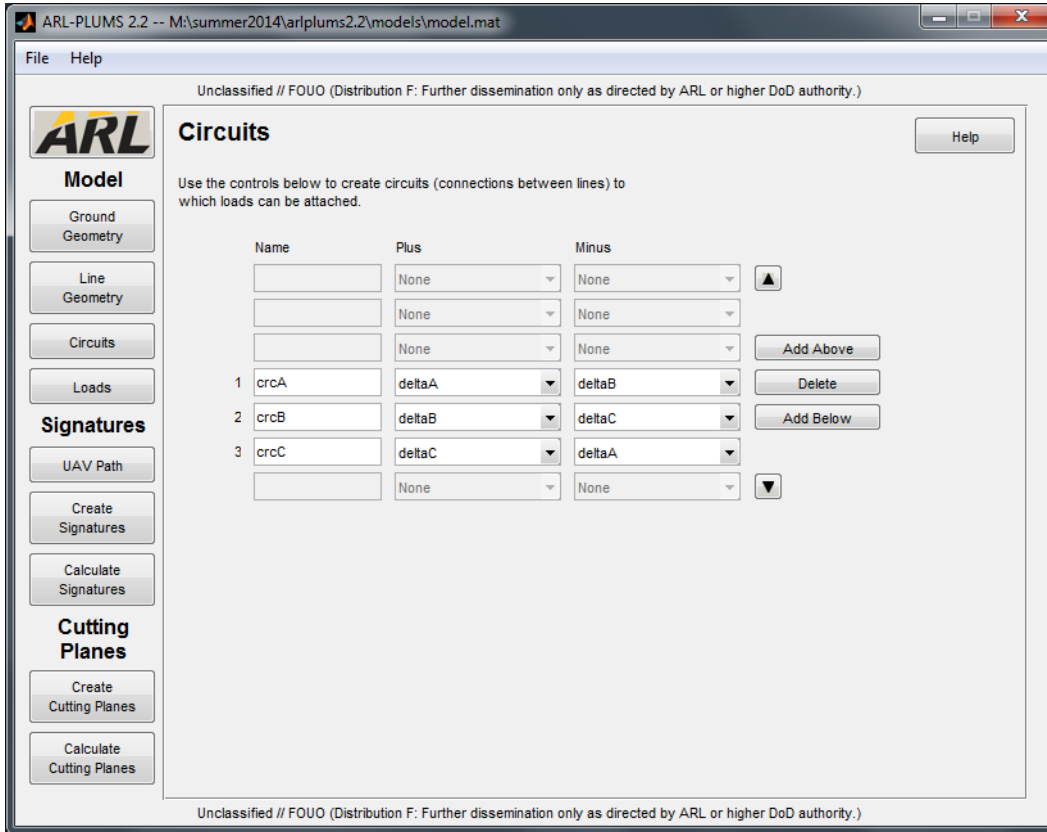


Fig. 30 The “Circuits” panel

7.4 Loads

The “Loads” panel (Fig. 31) is where one attaches incremental loads to the circuits. These incremental loads are added to the base load specified in the line geometry panel (see Section 7.2). Use the up and down arrows to scroll through the loads; click “Delete” to delete the load in the middle row; and click “Add Above” or “Add Below” to add a load above or below the load in the middle row, respectively. Give each load a name, choose a type (in version 2.2, the only type available is “Simple”, which, depending on the power factor, is either a resistive load or a partially inductive load), enter the power and power factor of the load, and choose which circuit to attach the load to from the drop-down box.

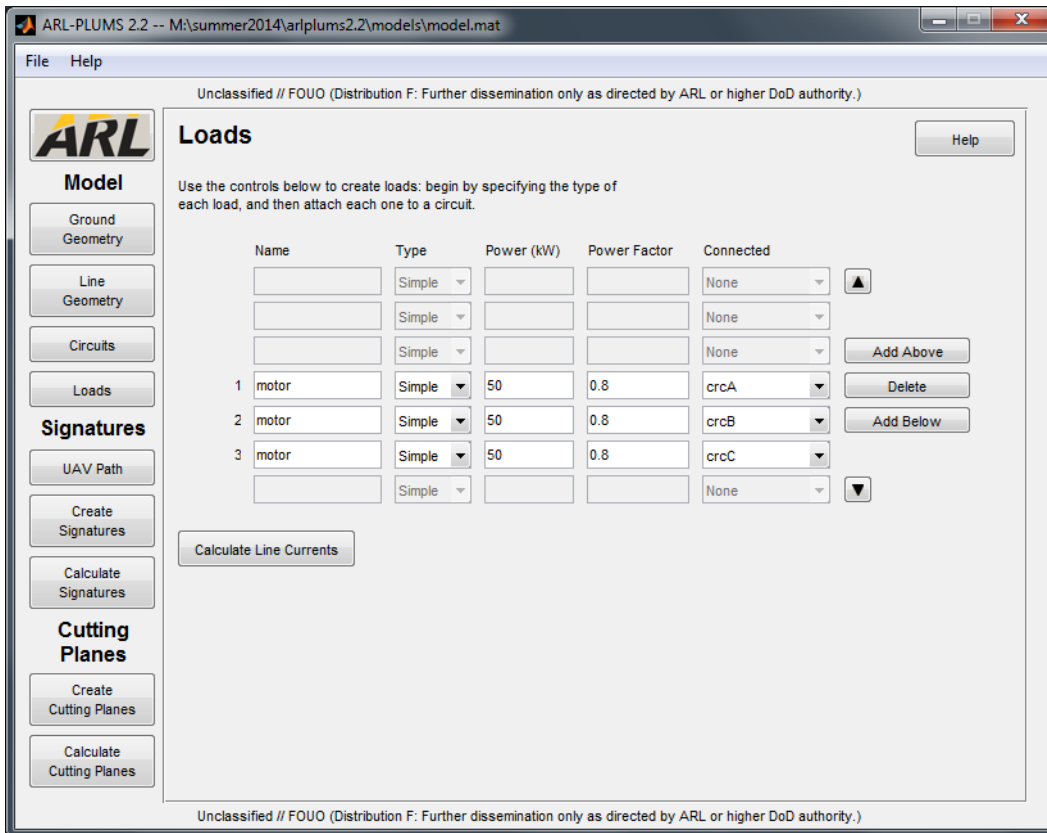


Fig. 31 The “Loads” panel

Click “Calculate Line Currents” to bring up a window showing the total current on each line (Fig. 32). The total current on each line is equal to the base current on each line that was entered in the “Line Geometry” panel plus the current due to the incremental loads.

Line	rms (A)	Phase (deg)
deltaA	10.7679	-3.00667
deltaB	10.7679	-123.007
deltaC	10.7679	116.993

Fig. 32 The window that appears when clicking “Calculate Line Currents”. These currents correspond to 10-A resistive base load and three 50-kW motors with a power factor of 0.8 being attached to the 115-kV 3-phase circuit seen in Fig. 24. In this particular example, the base load is $(115 \text{ kV}) * (10 \text{ A}) * \text{sqrt}(3) = 2 \text{ MW}$, which easily dwarfs the three 50-kW motors.

To compute how much current is added by attaching a load,

$$I_{\text{rms}} = \frac{P}{V_{\text{rms}}F} \tag{10}$$

$$I_{\text{phase}} = V_{\text{phase}} - \arccos(F) \tag{11}$$

where V_{rms} and V_{phase} are the rms magnitude and phase of the voltage across the positive and negative terminals of the circuit to which the load is being attached, and P and F are the power and power factor of the load, respectively. This calculator can be used iteratively to compute the total currents (magnitude and phase) on each line due to several different loads.

7.5 Modeling DC Power Lines

ARL-PLUMS is capable of modeling DC power lines. To do this, enter “0” for the operating frequency. Because ARL-PLUMS was designed for AC frequencies, many of the calculations performed internally assume a nonzero frequency. Thus, when modeling the DC case, one should take care when interpreting the results. First of all, all values are completely real, so a number can only be positive or negative. Thus, phase information is mostly useless: a positive number’s phase will be 0° and a negative number’s phase will be -180° . Second, the 3-phase circuits normally available in the “Line Geometry” panel cannot be used because they automatically introduce phase rotations, which do not exist in the DC case. Third, in all of the plots, rms magnitudes are plotted, not absolute magnitudes. In the AC case, there is a factor of 1.414 (the square-root of 2) between these. In the DC case, they are equal. Internally, ARL-PLUMS assumes the AC case, so rms magnitudes

will be low by a factor of 1.414 in the DC case. Again, be careful when interpreting results.

8. Construct the UAV Path and Calculate Signatures

8.1 UAV Path

The “UAV Path” panel (Fig. 33) is where one constructs the UAV path. This is done by creating a sequence of control points through which the UAV path will pass. The interpolation routine uses cubic splines, which are smooth (the path is twice differentiable) and the path always goes through the control points. Use the up and down arrows to scroll through the control points. To add a control point above or below the control point in the middle row, click “Add Above” or “Add Below”, respectively. To delete the control point in the middle row, click “Delete”. At a minimum, there needs to always be at least 2 control points, so when there are only 2 control points, one cannot delete either of them.

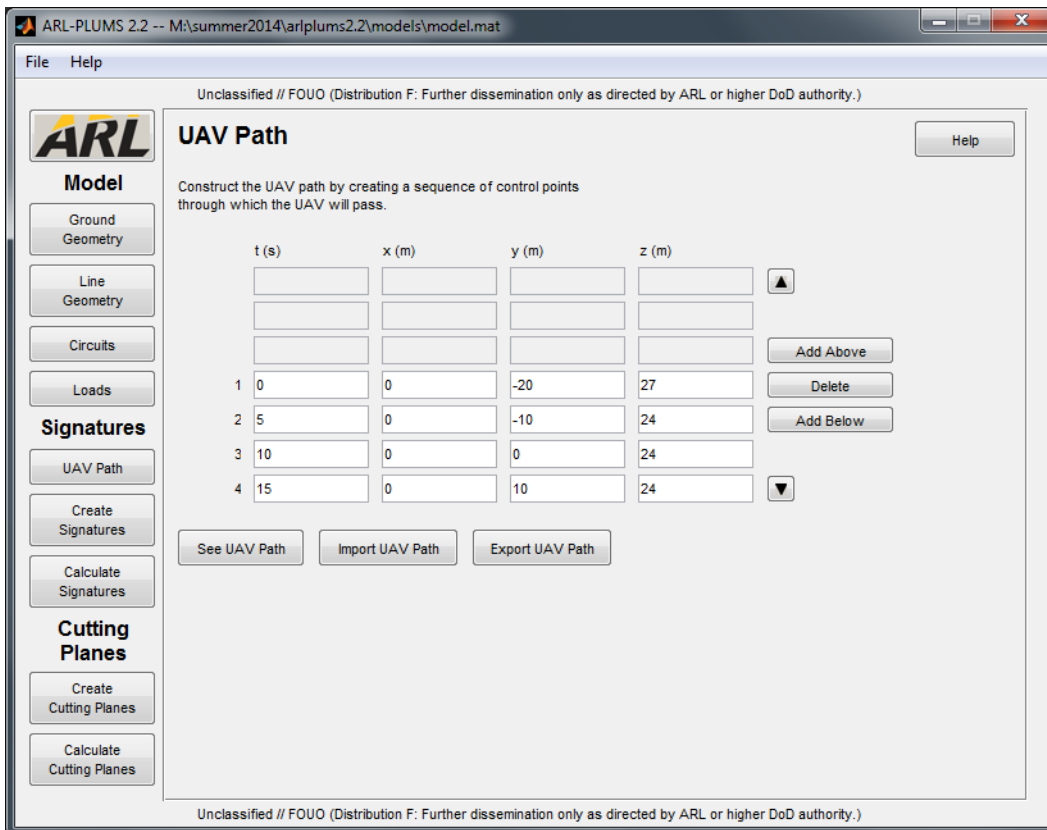


Fig. 33 The “UAV Path” panel

To see the UAV path, click “See UAV Path”. A window will appear showing the UAV path along with the ground and line geometries (Fig. 34). This window

updates as one adds, deletes, and modifies the control points. One can import a UAV path by clicking “Import UAV Path”. Likewise, one can export a UAV path by clicking “Export UAV Path”. The imported and exported UAV paths are stored using a simple CSV format.

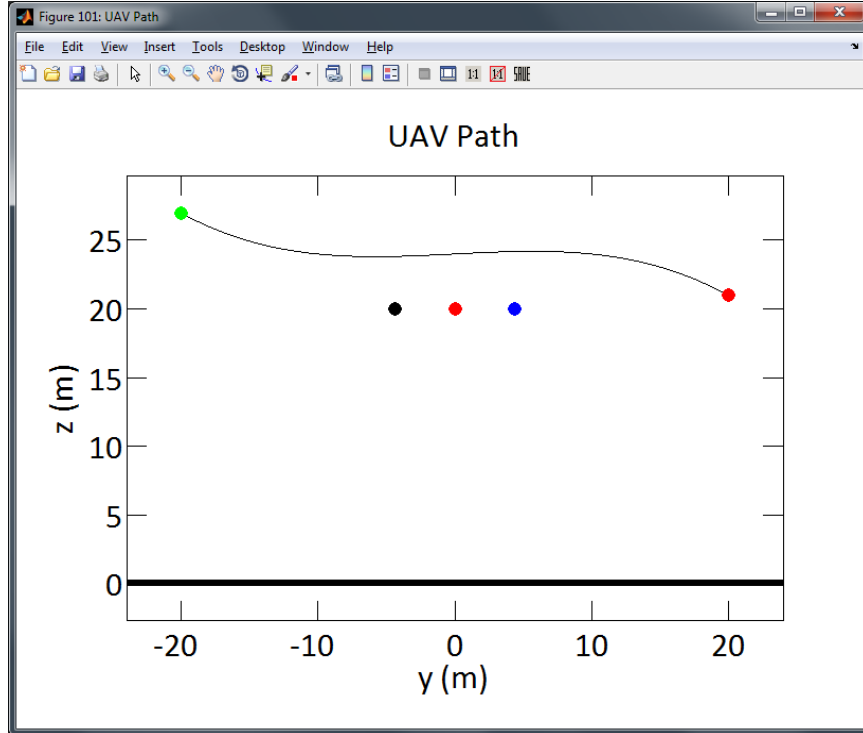


Fig. 34 The window that appears when one clicks “See UAV Path”

8.2 Create Signatures

The “Create Signatures” panel (Fig. 35) is where one enters the parameters for calculating the signatures. First, choose how the signatures will be sampled. One can either enter the total number of samples or the sampling rate (the number of samples per second). Use the drop-down box to select which of these to use. Second, one can choose how much noise to add to the signatures. The noise is Gaussian white noise with independent and identically distributed samples. The values entered are the standard deviations of these samples. When one changes the values in E_y and E_z , the value in E changes automatically, so that

$$E = \sqrt{E_y^2 + E_z^2} \quad (12)$$

Conversely, when one updates the value in E , the values in E_y and E_z update automatically, so that

$$E_y = \frac{E}{\sqrt{2}} \qquad E_z = \frac{E}{\sqrt{2}}$$

(13)

The same thing happens for H_y , H_z , and H .

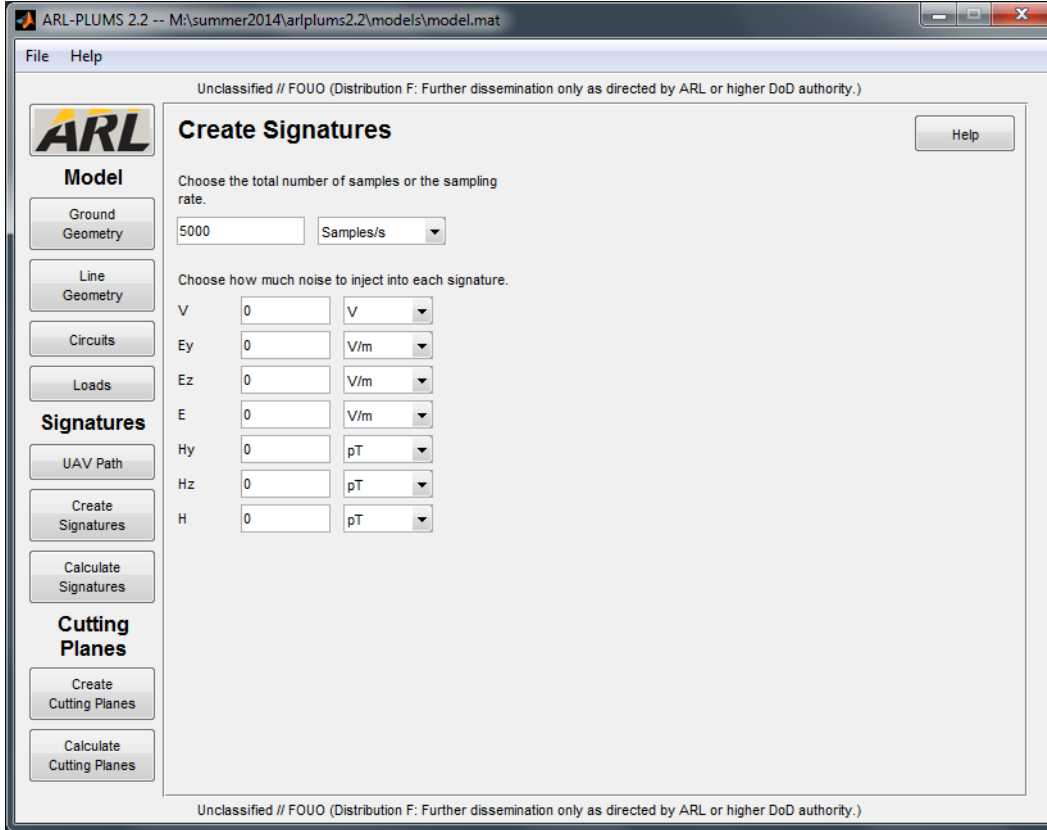


Fig. 35 The “Create Signatures” panel

8.3 Calculate Signatures

The “Calculate Signatures” panel (Fig. 36) is where one calculates the signatures. To begin, select the units to use when plotting and saving the signatures. Then, choose which signatures to calculate and plot, and click “Calculate Signatures”. A collection of windows will appear showing the signatures: Fig. 37 shows the signatures for “ E_y , rms”, “ E_y , phase”, and “ $E_y(t)$ ” with no noise added, and Fig. 38 shows the same signature for “ $E_y(t)$ ”, but with noise added. The signatures, “ E , rms” and “ H , rms”, are computed using

$$X_{\text{rms}} = \sqrt{X_{y,\text{rms}}^2 + X_{z,\text{rms}}^2} \qquad (14)$$

where X is replaced by E or H .

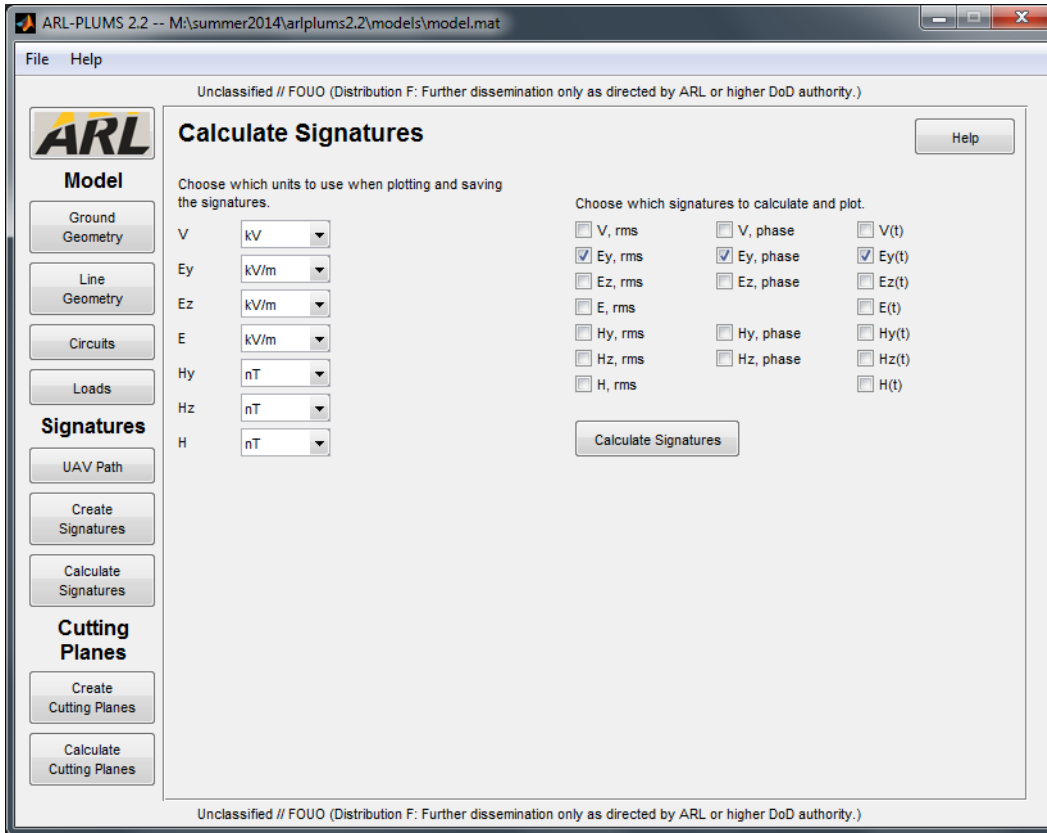


Fig. 36 The “Calculate Signatures” panel

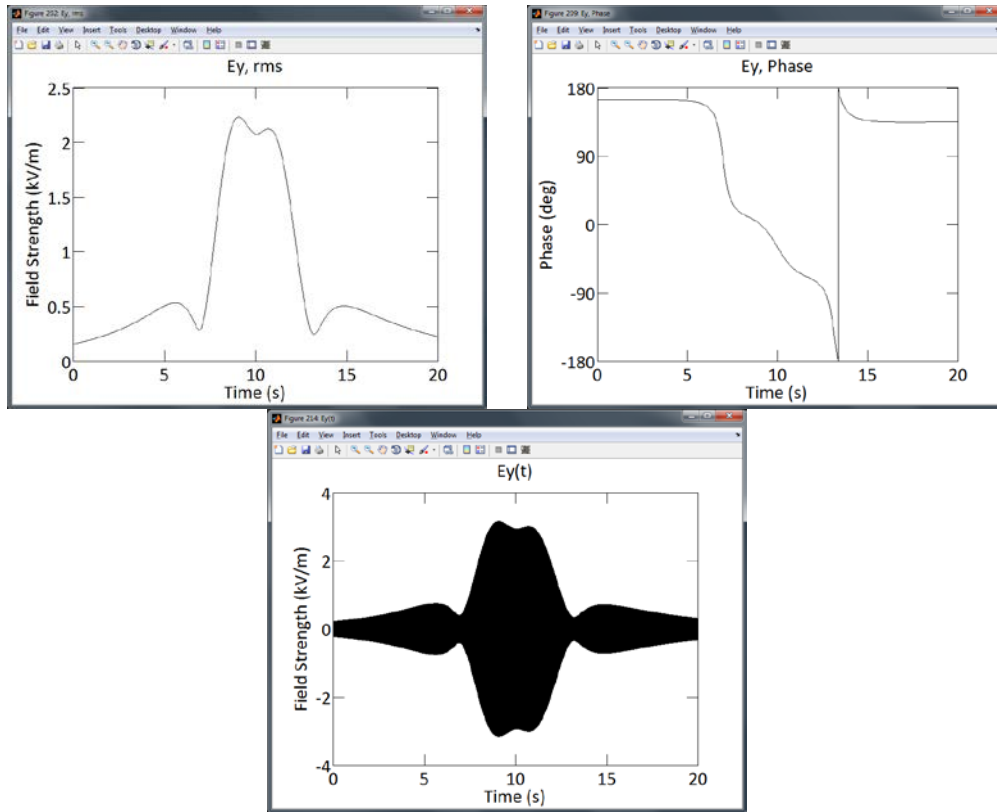


Fig. 37 The signatures for “ E_y , rms”, “ E_y , phase”, and “ $E_y(t)$ ” with no noise added

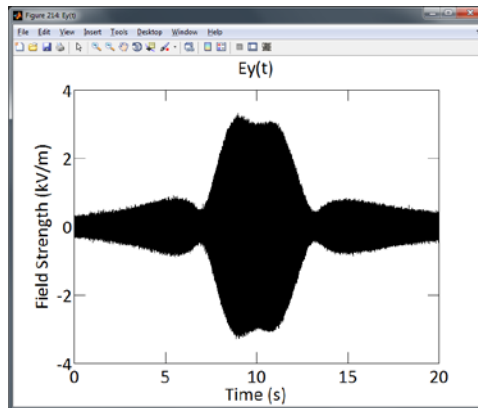


Fig. 38 The signature for “ $E_y(t)$ ” with noise added

9. Create and Calculate Cutting Planes

9.1 Create Cutting Planes

The “Create Cutting Planes” and “Calculate Cutting Planes” panels (Figs. 39 and 41) allow one to better visualize the electric and magnetic fields around the lines.

A cutting plane is a rectangle perpendicular to the lines over which one can compute and plot V , E_y , E_z , E , H_y , H_z , or H .

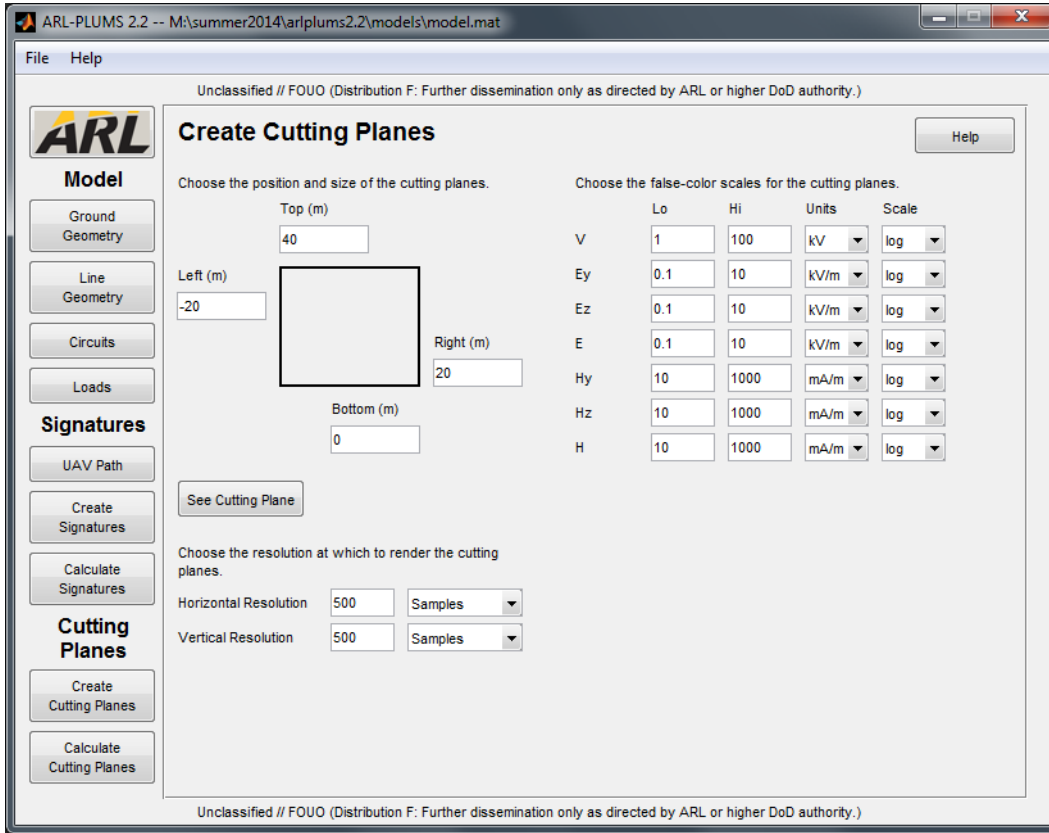


Fig. 39 The “Create Cutting Planes” panel

Every cutting plane is perpendicular to the lines, but one can set the size (left, right, top, and bottom extents) and resolution (the number of points at which the field quantities are calculated along the cutting plane). To set the extents of the cutting plane, enter a value for “Left”, “Right”, “Top”, and “Bottom”. To see the dimensions of the cutting plane, click “See Cutting Plane”. This will bring up a window showing the cutting plane along with the ground and line geometries and UAV path (Fig. 40). This window updates as one changes the parameters. Choose the resolution of the cutting plane in the horizontal and vertical directions by either choosing the total number of samples or the number of samples per meter. One should aim for a total number of samples of around 300 in each direction for high-quality plots. The higher the resolution, the better the plots will look, but the longer they will take to compute. The cutting planes are rendered using a false-color scale. One can choose the ranges and types (e.g., linear or logarithmic) of scales used, as well as which units to use when plotting.

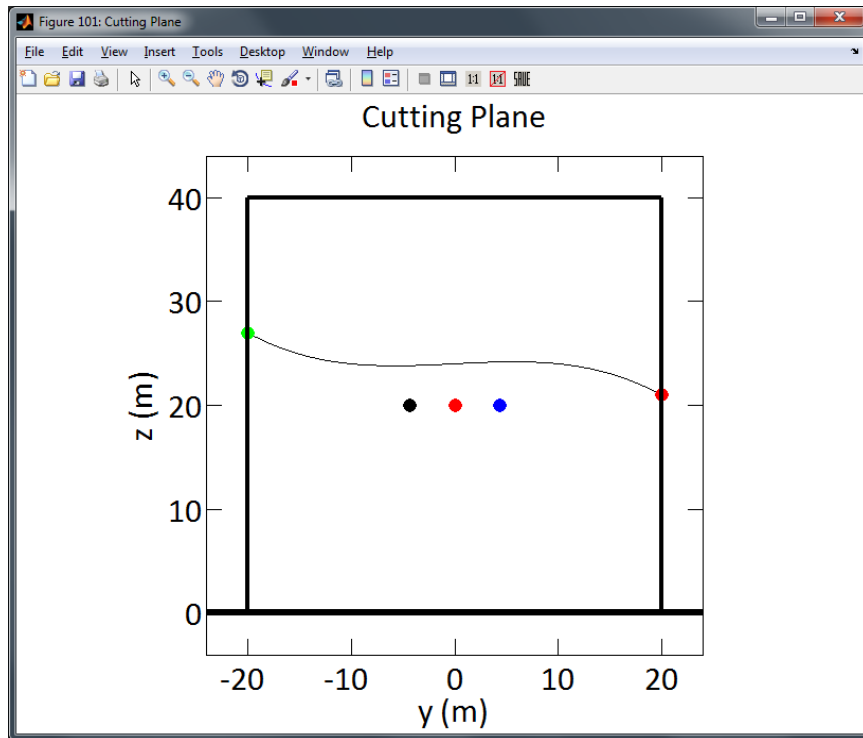


Fig. 40 The window that appears when one clicks “See Cutting Planes”

9.2 Calculate Cutting Planes

The “Calculate Cutting Planes” panel (Fig. 41) is where one calculates the cutting planes. Choose which cutting planes to calculate and plot, and click “Calculate Cutting Planes”. A window will appear for each cutting plane selected. Figure 42 shows the cutting planes for “ E_y , rms” and “ E_y , phase”.

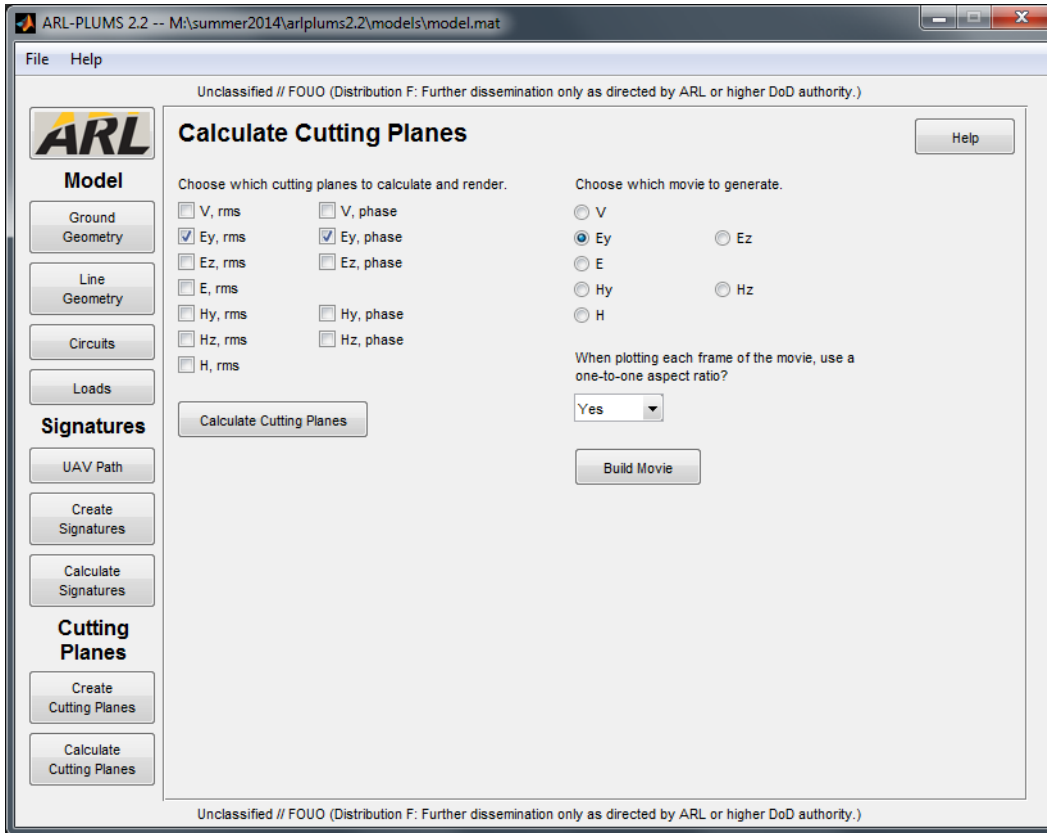


Fig. 41 The “Calculate Cutting Planes” panel

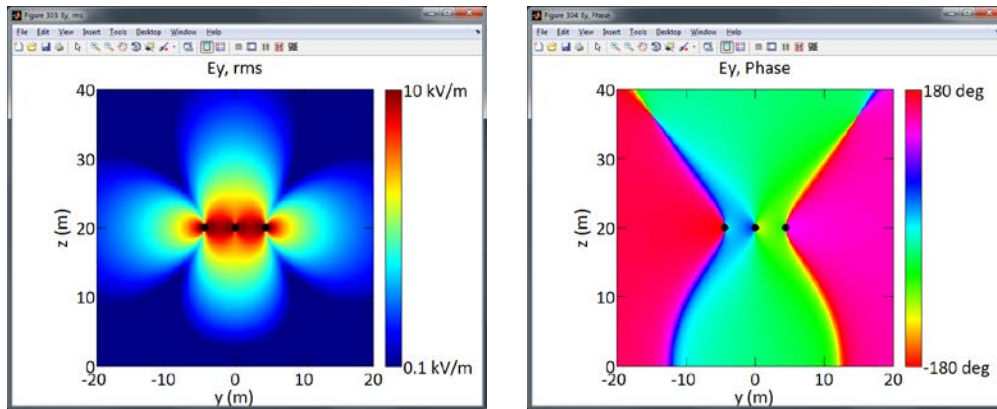


Fig. 42 The cutting planes for “ E_y , rms” and “ E_y , phase”

One can also generate movies, which show V , E_y , E_z , E , H_y , H_z , or H along a cutting plane over 1 complete AC cycle. To do this, choose which field quantity to use, select whether a 1-to-1 aspect ratio is wanted when rendering the frames, and click “Create Movie”. A sequence of frames will be plotted, saved, and combined to form a movie (Fig. 43). The movie is exported as an uncompressed AVI file, which can then be imported into a third-party application and compressed into a more manageable size.

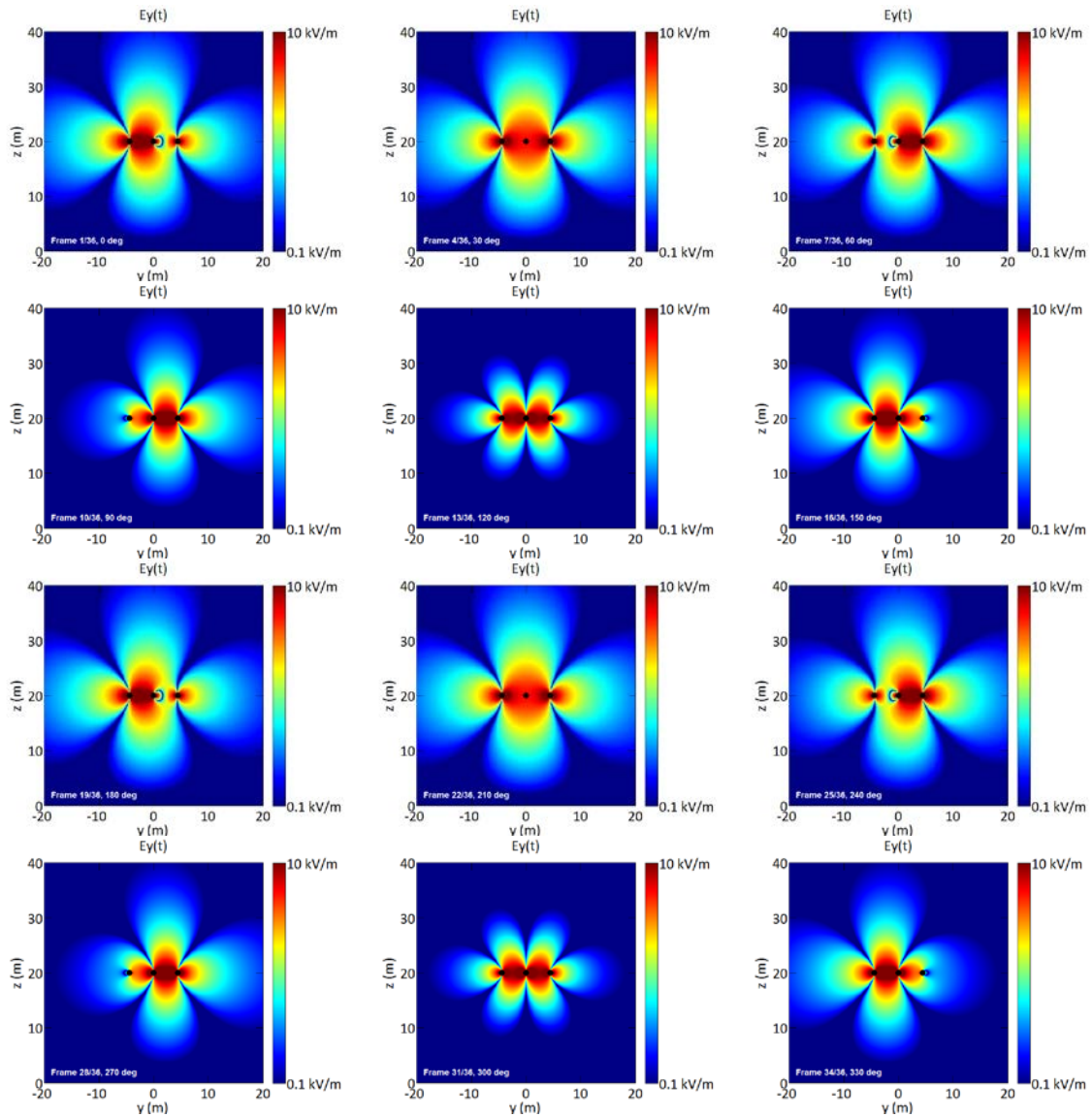


Fig. 43 Twelve frames of an example movie

10. Accessing ARL-PLUMS Functionality in MATLAB

Sections 10–11 describe how to operate ARL-PLUMS at the API level. These sections form a basic user manual for those who want to use ARL-PLUMS as a MATLAB toolbox.

The ARL-PLUMS API contains ten functions. Table 1 lists these functions, gives a short description of each, and says which section describe each in more detail. These functions provide access to some of the functionality in ARL-PLUMS, including the compute engine and 2 plotting functions. There is a separate P-File for each function in the API.

Table 1 The 10 functions available in the ARL-PLUMS API

Function	Description	Section
<code>apa_check_license</code>	Check that the copy of ARL-PLUMS being used has been properly licensed.	10.1
<code>apa_open_model</code>	Open a model.	10.2
<code>apa_check_model</code>	Check that the model is valid.	10.3
<code>apa_calculate_C</code>	Compute the capacitance matrix that describes the coupling between the lines.	10.4
<code>apa_calculate_line_lambda</code>	Compute the linear charge densities on the lines.	10.5
<code>apa_calculate_fields</code>	Compute the electric potential and electric and magnetic fields around the lines.	10.6
<code>apa_calculate_basis_fields</code>	Compute the basis fields, one for each line and field modality.	10.7
<code>apa_calculate_time_varying_fields</code>	Use the basis fields to compute time-varying fields given time-varying voltages and currents on the lines.	10.8
<code>apa_plot_signature</code>	Plot a time-domain signature.	10.9
<code>apa_plot_cutting_plane</code>	Plot a 2-D false-color cutting plane.	10.10

10.1 Check the ARL-PLUMS License

Internally, every function checks to make sure that the copy of ARL-PLUMS being used has been properly licensed. So that every function can perform this check, one needs to pass in a string containing the path to the directory in which ARL-PLUMS was installed. For example, consider the following example code fragment:

```
model = apa_open_model('.', 'models\model.mat');
```

The ARL-PLUMS directory (in this case, '.', which represents the current directory) was passed in before the other argument. After calling a function for the first time, that function will remember whether or not the license is valid and skip performing the check in subsequent calls. Each function remembers independently from the others. Even after a function has checked and remembered that the license is valid, the first argument is still reserved for this path. At this point, one can just pass in [].

To have every function in the API internally check and remember whether the ARL-PLUMS license is valid, call the following:

```
apa_check_license(arl_plums_dir);
```

where `arl_plums_dir` is a string containing the path to the directory in which ARL-PLUMS was installed (in the above example, this was '.'). After making

this call, one can pass in [] for arl_plums_dir (the first argument) in any subsequent call to the API.

10.2 Open an ARL-PLUMS Model

To open an ARL-PLUMS model, call

```
model = apa_open_model(arl_plums_dir, name);
```

where name is a string containing the path to the model one wants to open. Alternatively, assuming apa_open_model or apa_check_license has already been called previously, one can call

```
model = apa_open_model([], name);
```

The structure, model, contains 9 substructures, each corresponding to a panel in ARL-PLUMS:

- 1) Ground Geometry: model.ggpd
- 2) Line Geometry: model.mgpd
- 3) Circuits: model.lcpd
- 4) Loads: model.lpd
- 5) UAV Path: model.uppd
- 6) Create Signatures: model.spd
- 7) Compute Signatures: model.cspd
- 8) Create Cutting Planes: model.crcppd
- 9) Compute Cutting Planes: model.cacppd

Only model.ggpd and model.mgpd are used by the API. Changing any of the other seven substructures will have no effect when using the API. They are there because this is the same model structure used internally by ARL-PLUMS, but since many of the features in ARL-PLUMS are not available in the API, they are not used.

The lines are stored in the cell array, model.mgpd.lines. The elements in this cell array are structures describing the lines. There is 1 structure per line. For example, one of the lines in an example model shipped with ARL-PLUMS looks like the following:

```
model = apa_open_model('.', 'models\model.mat');  
model.mgpd.lines{2} =  
    enabled: 1  
    name: 'deltaB'
```



```
y: 0
z: 20
a: 0.0100
V: -4.6949e+04 - 8.1317e+04i
I: -8.2953 -12.7704i
color: 'Red'
```

The entries, `model.mgpd.lines{2}.V` and `model.mgpd.lines{2}.I`, are the voltage and current on this line, respectively.

10.3 Validate the Model

The API provides some simple sanity checks to make sure that the model is valid. To perform these checks, call the following:

```
status = apa_check_model(arl_plums_dir, model);
```

where

- `status = 0` when the model is valid;
- `status = 1` when there are no lines, enabled or disabled, in the model;
- `status = 2` when none of the lines in the model are enabled; and
- `status = 3` when 1 or more of the lines lie outside of the problem domain.

10.4 Compute Capacitances

As discussed previously, the voltages on the lines give rise to an electric field around the lines, and the currents on the lines give rise to a magnetic field around the lines. However, as discussed in Section 2, while the currents are the sources of the magnetic field, the voltages do not directly source the electric field. Instead, through capacitive coupling, the voltages induce linear charge densities on the lines, and these linear charge densities are the actual sources of the electric field. The capacitive coupling is described by the capacitance matrix, which can be computed using the following:

```
C = apa_calculate_C(arl_plums_dir, model);
```

10.5 Compute Linear Charge Densities

Similarly, the linear charge densities can be computed using the following:

```
line_lambda = apa_calculate_line_lambda(arl_plums_dir, model);
```

10.6 Compute Fields

To compute the electric potential and electric and magnetic fields (collectively called the fields) at a point in space around the power lines, call the following:

```
fields = apa_calculate_fields(arl_plums_dir, model, y, z);
```

where y and z are row vectors containing the positions of the evaluation points, and `fields` is a structure containing V , E_y , E_z , H_y , and H_z . As discussed previously, because ARL-PLUMS is a 2-D model, there is symmetry along the x axis, so there is no need to pass in x . Likewise, the electric and magnetic fields along the direction of the lines (i.e., in the x direction) are zero, so E_x and H_x are zero, so there is no need to return them.

Computing the fields in an example model (the same one from Section 10.2) shipped with ARL-PLUMS at 3 points on the ground, directly beneath the lines (i.e., $y = 0, 1$, and $2, z = 0$) yields the following:

```
fields = apa_calculate_fields('.', model, [0.0, 1.0, 2.0], [0.0,
0.0, 0.0]);
fields =
  V: [0 0 0]
  Ey: [0 0 0]
  Ez: [-5.7781 -10.0079i 35.3436 -34.7288i 74.5820 -60.2462i]
  Hy: [-0.0030 - 0.0047i -0.0065 - 0.0023i -0.0098 + 0.0001i]
  Hz: [0.0369 - 0.0240i 0.0362 - 0.0245i 0.0351 - 0.0247i]
```

In this particular example, $V = 0$ and $E_y = 0$ because the ground is an equipotential surface at zero potential.

The fields returned by this function depend on the geometry of the lines and the voltages and currents on those lines. One can modify these values in the model structure before calling this function. This is useful for simulating fields due to changing line geometry (e.g., wire sag due to heating, swaying lines due to wind) or currents that are changing over time. However, for the latter scenario, when the currents are changing, the API comes with more efficient means for doing this by providing functions for computing basis and time-varying fields.

10.7 Compute Basis Fields

The electric field at a point in space around the lines is the sum of the constituent electric fields due to each of the lines separately (Adelman 2009). For example,

$$\mathbf{E} = V_A \mathbf{E}_A + V_B \mathbf{E}_B + V_C \mathbf{E}_C \quad (15)$$

where, in this case, there are 3 lines, A, B, and C, their voltages are V_A , V_B , and V_C , and their constituent electric fields are $V_A \mathbf{E}_A$, $V_B \mathbf{E}_B$, and $V_C \mathbf{E}_C$. The constituent

electric field for line A, $V_A \mathbf{E}_A$, is computed by setting the voltage on line A to V_A , setting the voltages on lines B and C to zero, and solving for the electric field. The same is done for the other constituent electric fields. Dividing each of the constituent electric fields by their respective voltages yields 3 basis electric fields, \mathbf{E}_A , \mathbf{E}_B , and \mathbf{E}_C . A basis electric field is the electric field due to one of the lines being energized to 1 V and all of the others being grounded (set to 0 V). Thus, the electric field due to an arbitrary choice of voltages on the lines is a linear combination of the basis electric fields with the line voltages as the coefficients.

Likewise, the magnetic field at a point in space around the power lines is the sum of the constituent magnetic fields due to each of the lines separately. While the voltages on the lines give rise (indirectly) to the electric field, the currents on the lines give rise (directly) to the magnetic field. For the same 3-phase power line,

$$\mathbf{H} = I_A \mathbf{H}_A + I_B \mathbf{H}_B + I_C \mathbf{H}_C \quad (16)$$

where I_A , I_B , and I_C are the currents on the lines and $I_A \mathbf{H}_A$, $I_B \mathbf{H}_B$, and $I_C \mathbf{H}_C$ are the constituent magnetic fields. The constituent magnetic field, \mathbf{H}_A , is computed by setting the current on line A to I_A , setting the currents on lines B and C to 0 A, and solving for the magnetic field. The same is done for the other constituent magnetic fields. Dividing each of the constituent magnetic fields by their respective currents yields 3 basis magnetic fields, \mathbf{H}_A , \mathbf{H}_B , and \mathbf{H}_C . A basis magnetic field is the magnetic field due to one of the lines having 1 A and all of the others having 0 A. Thus, the magnetic field due to an arbitrary choice of currents on the lines is a linear combination of the basis magnetic fields with the line currents as the coefficients.

Another way to think about the basis electric and magnetic fields is that they behave like the impulse response of the electric and magnetic fields around the power lines.

To compute the basis fields, call the following:

```
basis_fields = apa_calculate_basis_fields(arl_plums_dir, model,
y, z);
```

The arguments are the same as for `apa_calculate_fields`. The return value looks similar as well, except each entry in `basis_fields` has more than one row:

```
basis_fields = apa_calculate_basis_fields('.', model, [0.0, 1.0,
2.0], [0.0, 0.0, 0.0]);
```

```
basis_fields =
    V: [3x3 double]
    Ey: [3x3 double]
    Ez: [3x3 double]
    By: [3x3 double]
    Bz: [3x3 double]
```

Each row corresponds to one of the lines. The basis fields for the line described by `model.mgpd.lines{k}` is the k th row. In this particular example, there are 3 rows since the model has 3 lines.

10.8 Compute Time-Varying Fields

As discussed in the previous section, one can linearly combine the rows in `basis_fields.V`, `basis_fields.Ey`, and `basis_fields.Ez` using the line voltages as the coefficients. Likewise, one can linearly combine the rows in `basis_fields.Hy` and `basis_fields.Hz` using the line currents as the coefficients. The API provides a function to do this for the user:

```
time_varying_fields =  
apa_calculate_time_varying_fields(arl_plums_dir, model, y, z,  
basis_fields, time_varying_line_V, time_varying_line_I);
```

The arguments, `time_varying_line_V` and `time_varying_line_I`, allow one to specify the voltages and currents on the lines as functions of time. Then, this function uses them and the basis fields to compute the fields at each time step due to the changing voltages and currents, and returns them in `time_varying_fields`. For example:

```
time_varying_fields = apa_calculate_time_varying_fields('.',  
model, [0.0, 1.0, 2.0], [0.0, 0.0, 0.0], basis_fields, rand(10,  
3), rand(10, 3));  
  
time_varying_fields =  
    V: [10x3 double]  
    Ey: [10x3 double]  
    Ez: [10x3 double]  
    By: [10x3 double]  
    Bz: [10x3 double]
```

The rows in `time_varying_fields` correspond to the timesteps, and the columns correspond to the evaluation points.

10.9 Plot a Signature

The API comes with 2 plotting functions, one for plotting signatures and one for plotting cutting planes. These are the same routines used internally by ARL-PLUMS.

To plot a signature, call the following:

```
f = apa_plot_signature(arl_plums_dir, k, name, t, s, measure,  
units);
```

where k is the figure number, `name` is the title of the plot, `t` is a row vector containing the times of the signature samples, `s` is a row vector containing the

values of the signature samples (should be the same length as t), `measure` is the quantity being plotted (e.g., 'Ey'), `units` are the units of the signature (e.g., 'V/m'), and `f` is the figure handle of the resulting plot. For example, calling the following code fragment will yield the left plot in Fig. 44:

```
f = apa_plot_signature('.', 1, 'Test Plot', 1 : 10, (1 : 10) .^
2, 'Ey', 'V/m');
```

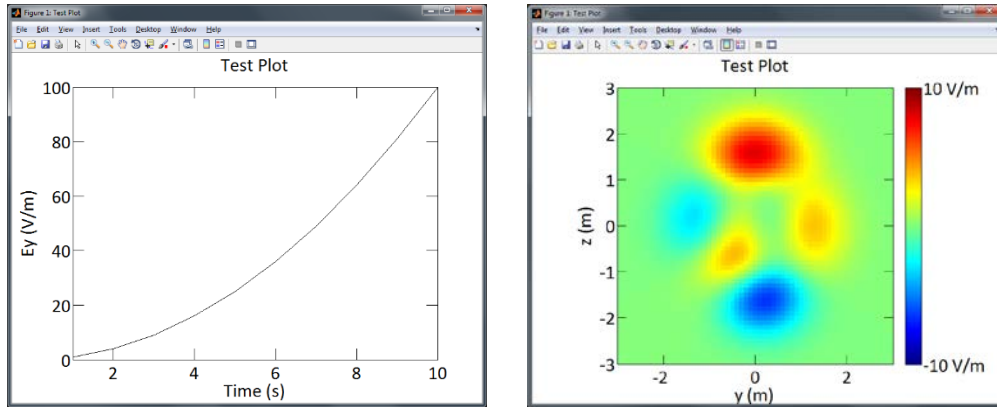


Fig. 44 Examples of using the 2 plotting functions

10.10 Plot a Field in a Cutting Plane

To plot a cutting plane, call the following:

```
f = apa_plot_cutting_plane(arl_plums_dir, k, model, name, y, z,
X, scale_un, scale_lo, scale_hi, scale_sc);
```

where `k` is the figure number, `model` is an ARL-PLUMS model (the line geometry is plotted on top of the cutting plane, use `[]` for no model), `name` is the title of the plot, `y` and `z` are the horizontal and vertical coordinates of the cutting plane samples (created using `meshgrid`), `X` are the values of the cutting plane samples, `scale_un` are the units of the cutting plane, `scale_lo` and `scale_hi` are the limits of the false-color scale, `scale_sc` is the type of false-color scale to use (the acceptable values are 'lin', 'log', or 'angle'), and `f` is the figure handle of the resulting plot. For example, calling the following code fragment will yield the right plot in Fig. 44:

```
[y, z] = meshgrid(-3 : 0.1 : 3, -3 : 0.1 : 3);
X = peaks(y, z);
f = apa_plot_cutting_plane('.', 1, [], 'Test Plot', y, z, X,
'V/m', -10, 10, 'lin');
```

For both of these plotting functions, the resulting figures can be modified after creating them by using any of the standard MATLAB graphics functions (e.g., `xlabel`).

11. Using the ARL-PLUMS API: ARL Wire-Avoidance Simulation Program (ARL-WASP)

Power lines pose a serious crash hazard to UAVs. This is because they are so widespread, hard to see, and strung at roughly the same height above the ground that UAVs fly. Thus, methods for detecting and avoiding them must be developed to prevent UAVs from crashing into them.

We have developed a novel method for detecting and avoiding power lines using the electric and magnetic fields they generate. Once avoided, our method also allows the UAV to use the lines as a navigational aid. Our method is divided into 2 phases: detection and avoidance. Both of these are described below.

The ARL Wire-Avoidance Simulation Program (ARL-WASP) allows a user to model a UAV flying around power lines and provides a test bench for rapidly testing these new methods. Like ARL-PLUMS, we developed ARL-WASP in MATLAB, and used the ARL-PLUMS API for simulating the electric and magnetic fields around the lines. Thus, in addition to demonstrating these new methods, ARL-WASP serves as a good example of using the ARL-PLUMS API.

11.1 Novel Power-Line Detection and Avoidance Methods

Our method is divided into 2 phases: detection, where the UAV measures the electric and magnetic fields to detect any nearby power lines; and avoidance, where the UAV steers away from the lines to avoid crashing into them.

11.1.1 Detection

Various combinations of the electric and magnetic fields, such as rms magnitudes, derivatives with respect to time and/or distance, ratios, and percent changes, can be used to detect the power lines. Each of these quantities has a specific structure around the lines, which can be exploited to determine how close to the lines the UAV is. For example, Fig. 45 shows the rms magnitude of the electric and magnetic fields around some example power lines. Notice how they form bull's-eye patterns around the lines. No matter the direction of approach, the UAV will measure larger and larger fields as it moves closer and closer to the lines.

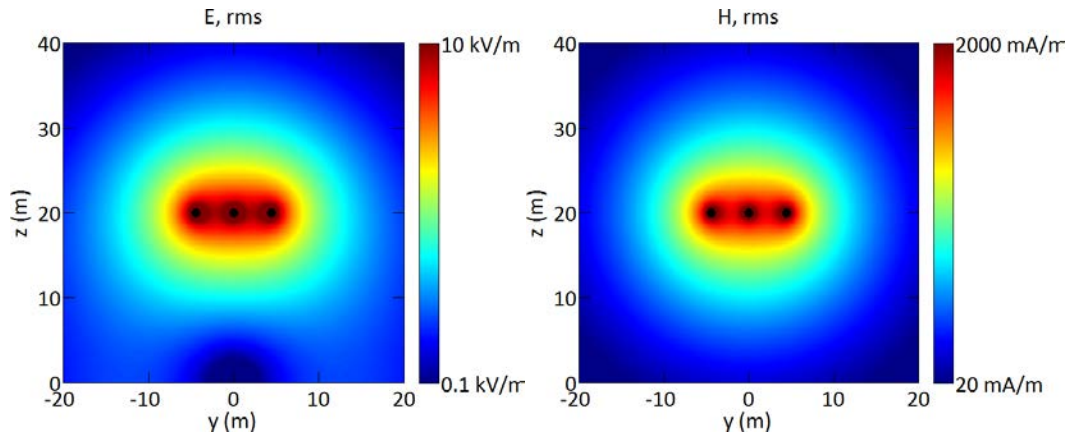


Fig. 45 The rms magnitude of the electric (left) and magnetic (right) fields around some example power lines. Notice how they form bull's-eye patterns around the lines.

One drawback to using the rms magnitudes of these fields is that they depend on the voltages and currents on the lines. The electric field will be much stronger near high-voltage transmission lines than near low-voltage distribution lines. Likewise, the magnetic field will be much stronger during peak energy usage in the afternoon on a hot day than at night since the lines will be carrying more current at those times. Thus, in addition to using the rms magnitude of these fields, in our method, we also look at their normalized spatial derivatives, both in the direction of flight and from side to side.

In total, up to 6 quantities are computed: the rms magnitude of the electric and magnetic fields; the percent change of these per unit of forward distance traveled by the UAV; and the percent change of these from side to side of the UAV.

During the detection phase, as soon as one of these quantities exceeds a specified threshold, the UAV detects the power lines.

11.1.2 Avoidance

Once the UAV has detected the lines, it must take action to avoid crashing into them. ARL-WASP currently includes 6 possible actions to take:

- 1) Turn one way and follow the lines downstream toward the load.
- 2) Turn the other way and follow the lines upstream away from the load, toward the power plant or generator.
- 3) Fly over the lines.
- 4) Fly over the lines and turn downstream.
- 5) Fly over the lines and turn upstream.

6) Turn around and go back the way it came.

For many of these actions, the UAV must know the orientation of the lines. Our method uses the time-averaged Poynting vector to determine this. Because the electric and magnetic fields are both perpendicular to the lines, and because the Poynting vector is the cross product of these fields, the time-averaged Poynting vector is parallel (or very close to parallel) to the lines. It is computed by (Jackson 1993)

$$\langle \mathbf{P} \rangle = \frac{1}{2} \text{Re}\{\mathbf{E} \times \mathbf{H}^*\} \quad (17)$$

where \mathbf{E} is the electric field, \times is the cross product, \mathbf{H} is the magnetic field, $*$ is the complex conjugate, $\text{Re}\{\}$ extracts the real component of a complex vector, and $\langle \mathbf{P} \rangle$ is the time-averaged Poynting vector.

11.2 ARL-WASP

One can demo our new wire-avoidance method using ARL-WASP. The ARL-WASP GUI is shown in Fig. 46. This is where one enters the simulation parameters. Once all of the simulation parameters have been entered, click “Simulate”. Three windows will appear (Fig. 48). The window on the left shows the ARL-PLUMS model (in this case, a single 3-phase power line), the UAV (the large black dot), and the path the UAV has taken (the black line trailing the black dot). The window on the top right shows the 6 quantities discussed earlier (only 4 if the 1-sensor setup was chosen). The rms magnitudes of the electric and magnetic fields are shown in the 2 left subplots, and the percent changes are shown in the 2 right subplots. The window on the bottom right shows the velocity vector (black) and the time-averaged Poynting vector (black). In this case, the UAV has already lined up its velocity vector with the time-averaged Poynting vector, so all one sees is the velocity vector. Some of the actions have multiple stages. For example, when “Fly Over and Turn Downstream” is chosen, there are 4 stages: 1) detect the lines, 2) pull up, 3) level off, and 4) turn downstream (Fig. 47). Whenever the algorithm proceeds from one stage to the next, a vertical purple dotted line will be plotted in all 4 subplots in the top-right window. This can be seen in Fig. 48.

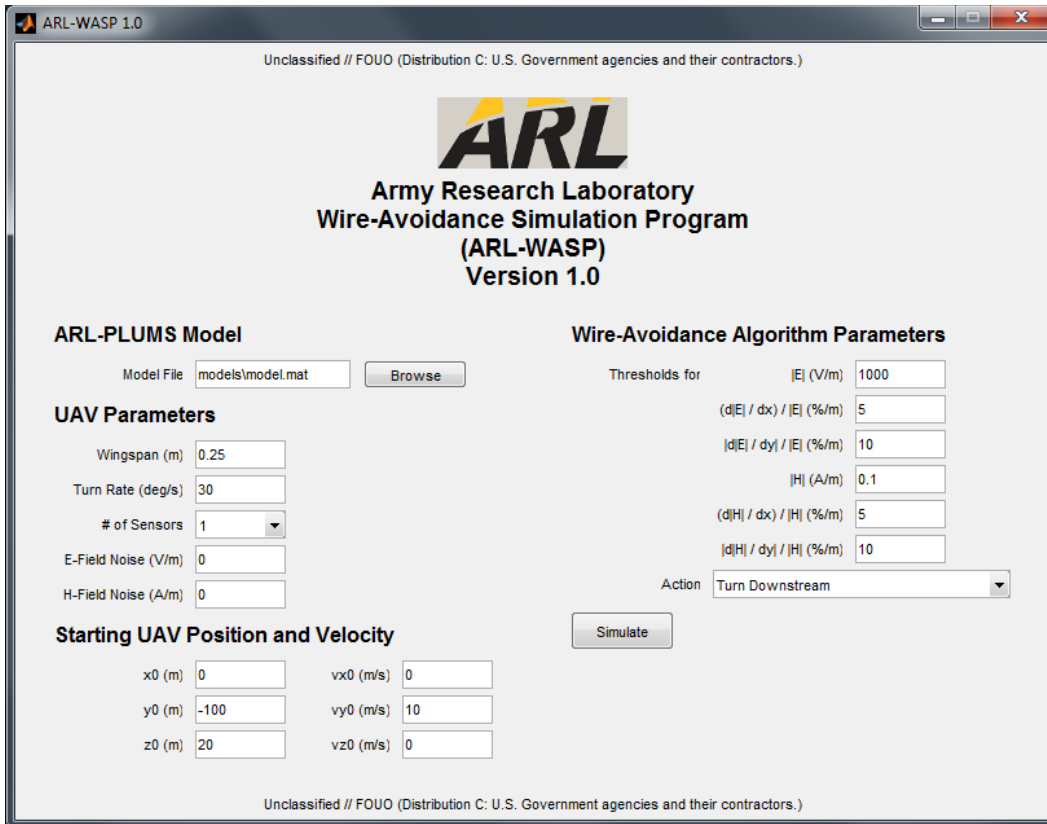


Fig. 46 The ARL-WASP GUI. This is where one enters the simulation parameters and start the simulation.

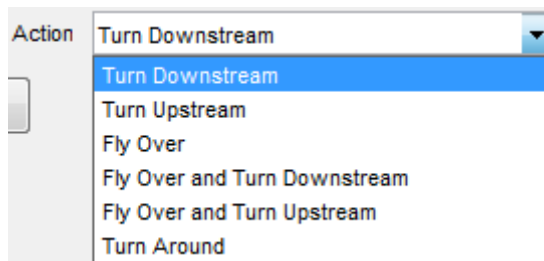


Fig. 47 The different wire-avoidance algorithms to choose from, which are the same as those described previously

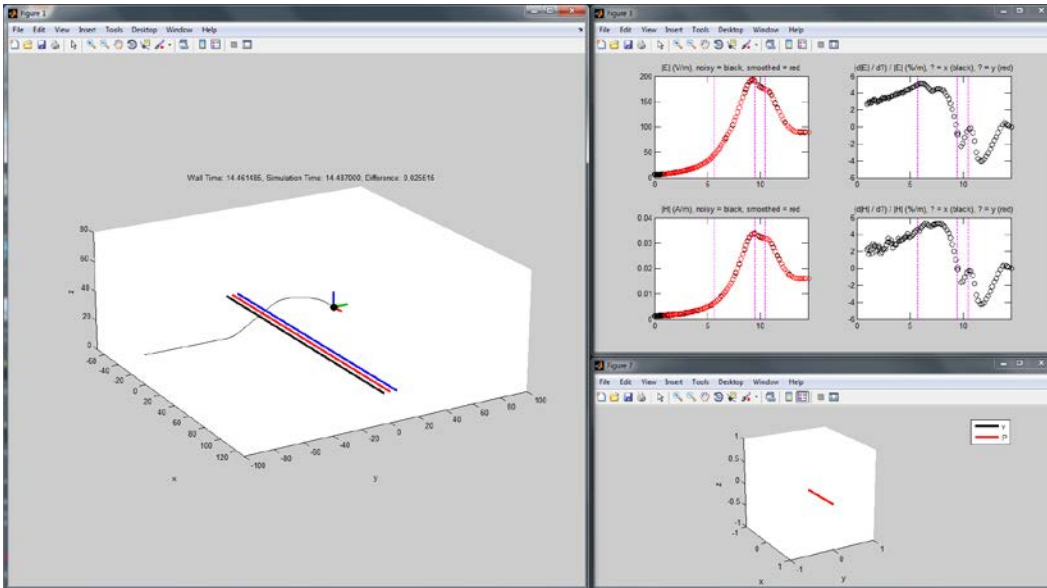


Fig. 48 The ARL-WASP simulation in action. In this case, “Fly Over and Turn Downstream” was chosen as the action to take.

Internally, ARL-WASP implements a 3-step simulation loop (Fig. 49):

1. **Measure the UAV’s Fields:** Measure the electric and magnetic fields at the UAV’s position, and use them to compute the 6 quantities (only 4 if the one-sensor setup is used) that are used for detecting the lines.
2. **Wire-Avoidance Algorithm:** Pass in these quantities to the multi-stage wire-avoidance algorithm to determine if the UAV should turn, and if so, in which direction.
3. **Flight Dynamics Model:** Pass the steering information returned by the wire-avoidance algorithm into the flight dynamics model, which moves the UAV according to that info.

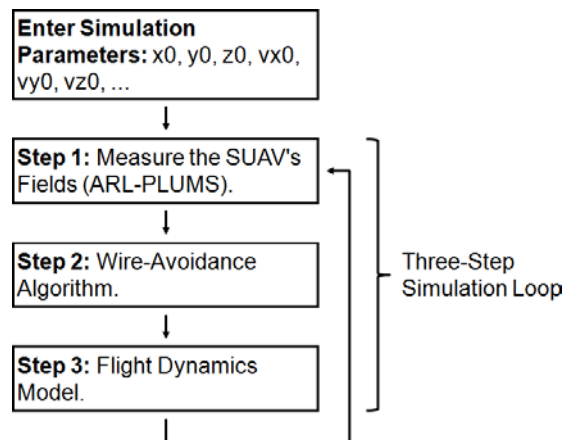


Fig. 49 The 3-step simulation loop used in ARL-WASP

The MATLAB code for the 3-step simulation loop described previously is shown here:

```
%
% Start the simulation.
%
function start_simulation()
    global running;

    uav_field_quantities = [];
    while (running)
        % Step 1: Measure the UAV's fields.
        uav_fields = measure_uav_fields();
        uav_field_quantities = calculate_uav_field_quantities( ...
            uav_field_quantities, uav_fields);

        % Step 2: Use the wire-avoidance algorithm to determine if
the UAV
        % should turn, and if so, in which direction.
        [ ...
            should_turn, direction, stage_changed ...
        ] = wire_avoidance_algorithm(uav_field_quantities);

        % Step 3: Update the position and velocity of the UAV using
the flight
        % dynamics model. Pass in the steering information returned
by the
        % wire-avoidance algorithm.
        flight_dynamics_model(should_turn, direction);

        % Update the simulation.
        update_simulation(uav_field_quantities, stage_changed);
    end
end
```

The MATLAB code for the `measure_uav_fields` functions, which actually makes use of the ARL-PLUMS API, is shown here:

```
%
% Measure the UAV's fields.
%
function uav_fields = measure_uav_fields()
    global model;
    global uav_x;

    uav_fields = apa_calculate_fields('.', model, uav_x(2),
uav_x(3));
    uav_fields.Ex = 0.0;
    uav_fields.Hx = 0.0;
end
```

12. Additional Information

12.1 Feedback

Feedback on ARL-PLUMS and/or the models and simulations shipped with it is welcome and encouraged. Please direct inquiries to the following:

US Army Research Laboratory
ATTN: RDRL-SES-P (David Hull)
2800 Powder Mill Road
Adelphi, MD 20783-1138

12.2 License Information

ARL-PLUMS has been developed by ARL for US Government purposes. A license to use this software may be granted as needed under the terms of an appropriate agreement with ARL (contract, CRADA, PLA, MOA, MOU, ISSA, etc.). Absent specific terms to the contrary in that agreement, this software may not be re-packaged, sold or otherwise re-distributed, or de-compiled or otherwise reverse-engineered.

12.3 Disclaimers

ARL-PLUMS is intended for research and development (R&D) purposes only. Suitability for particular applications is not implied, and the accuracy of specific results is not guaranteed.

12.4 Fair Use

If results from ARL-PLUMS are used in research products (e.g., a paper or presentation), the assistance of ARL should be acknowledged; use of the ARL logo with ARL-PLUMS figures is sufficient for this purpose.

13. References

- Adelman R, Hull D. Basis models for complex electrostatic simulations. Proceedings of the Electrostatics Joint Conference, Boston, MA, 2009.
- Bergen AR. Power systems analysis. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1986, pp. 100–115.
- Blalock Thomas J. The first polyphase system: a look back at two-phase power for AC distribution. in IEEE Power and Energy Magazine; March–April 2004, ISSN 1540–7977; pp. 63–66.
- Cheng DK. Field and wave electromagnetics, Second Edition, Addison-Wesley Publishing Company, 1992, pg. 129.
- Harrington R. Matrix methods for field problems. Proc. IEEE. Feb 1967; 55(2):136–149.
- Hull DM. Time-varying electrostatic modeling techniques. Proceedings of the 1997 ARL Sensors and Electron Devices Symposium, pp. 209–212, College Park, MD, 14–15 January, 1997.
- Hull D, Adelman R. An interactive 2-D power-line modeling and simulation tool. Proc. SPIE 8382. 2012;838204.
- Jackson JD. Classical electrodynamics, Third Edition, Hamilton Printing Company, 1999, pg. 43.
- Jeans J. The Mathematical Theory of Electricity and Magnetism, Fifth Edition, Cambridge University Press, 1951, pg. 206.
- Lambdin DL. A comparison of measurement techniques to determine electric fields and magnetic flux under EHV overhead power transmission lines, ORP/EAD US Environmental Protection Agency, March, 1978. Technical Note 78-1.
- MathWorks, 2014. MATLAB and simulink for technical computing. <http://www.mathworks.com/>.
- Olsen RG, Wong PS. Characteristics of low frequency electric and magnetic fields in the vicinity of electric power lines. IEEE Transactions on Power Delivery. October 1992;7(4):2046–2055.
- Woodson HH, Melcher JR. Electromechanical dynamics, John Wiley and Sons, Inc., New York, 1968, pp. B1–B38.

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL SES P
R ADELMAN
D HULL