ARL-TR-9967 • Sep 2024

Data Processing and Visualization System for the Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM)

by Leelinda P Dawson and Yansen Wang

NOTICES

Disclaimers

ARL-TR-9967 ● SEP 2024

Data Processing and Visualization System for the Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM)

Leelinda P Dawson and Yansen Wang
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED: START DATE	END DATE
September 2024	Technical Report	10/01/2023	8/31/2024

4. TITLE AND SUBTITLE
Data Processing and Visualization System for the Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Leelinda P Dawson and Yansen Wang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-NA Adelphi, MD 20783	ARL-TR-9967

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
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCIDs: Leelinda P Dawson, 0000-0003-4209-8459; Yansen Wang, 0000-0003-1273-4290

14. ABSTRACT
The Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM) is the next-generation microscale atmospheric model developed by the US Army Combat Capabilities Development Command Army Research Laboratory to simulate and predict the atmospheric boundary layer's turbulent wind, temperature, and other scalar transports over complex terrain and urban areas. ABLE-LBM requires grid-rasterized terrain elevation, building, and vegetation data for the surface boundary information before the model can be used. This report details the development of a Google Geographical Information System (GIS)-based graphical user interface web application for ABLE-LBM that performs data preprocessing, postprocessing, and graphic visualization. This system takes advantage of the numerous resources in map and image data retrieving/handling and overlaying, as well as other GIS features offered by the Google Maps/Earth platform, which improves the system's functionality and ease of use.

15. SUBJECT TERMS
microscale atmospheric boundary layer modeling; terrain and urban data preprocessing; GIS application for atmospheric modeling; Network, Cyber, and Computational Sciences

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	23

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Leelinda P Dawson	(301) 394-5636

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

The Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM)[1–4] is the next-generation microscale atmospheric model developed by the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL). It simulates turbulent atmospheric boundary layer wind, temperature, moisture, and aerosol transports over complex terrain and urban areas. The model results can be applied for unmanned aircraft system flight control, acoustic/optical wave propagations, and transport and dispersion of materials in the atmospheric boundary layer.

ABLE-LBM requires grid-rasterized terrain elevation, building, and vegetation data for the surface boundary information before the model can be used to predict atmospheric variables that are affected by the surface features. This report details the development of a Google Geographical Information System (GIS)-based graphical user interface (GUI) web application for ABLE-LBM, which involves preprocessing the input data into the model and postprocessing its output data after running the model. It also explains the development of a graphic visualization system to display the ABLE-LBM's 2- or 3-D atmospheric variables on a Linux computer via Google Maps and Google Earth. This allows the user to easily visualize and understand the output from ABLE-LBM to assist in decision-making.

2. ABLE-LBM Web Application Development

The ABLE-LBM web application uses Node.js,[5] an open-source JavaScript runtime environment, as the middleware software on a localhost server, which can be launched on a Linux computer as described in Dawson.[6] An advantage of Node.js implementation is that it provides a secure method to run the executables over the Internet. The development for the ABLE-LBM web application involved data preprocessing, running the model, data postprocessing, and graphic visualization. The data preprocessing methods described in Dawson[2,6] were used with some minor changes. After running ABLE-LBM, the model's output data is postprocessed using a Java-based executable. A graphic visualization system for viewing the ABLE-LBM output data on the web application via Google Maps and Earth was also developed.

Figure 1 shows the overall execution and data flow of the design implementation for the ABLE-LBM web application. The main ABLE-LBM Node.js web application is "ABLE-LBM.js," which makes calls to other built-in application modules that were developed and implemented based on the processing requirements of ABLE-LBM. The Embedded JavaScript (EJS) modules, "ABLE-

LBM.ejs," "domain_para.ejs," and "generate_bldg.ejs," handle the data preprocessing for ABLE-LBM. They are stored in the "AppData/views/pages" subdirectory of the application's software package. A Java-based executable called "ABLE_LBM_Postproc" was also developed to handle reading the ABLE-LBM binary output data after running the model to produce the postprocessing data needed as input into the graphic visualization system. The EJS module "display_wind.ejs" handles reading the ABLE-LBM postprocessing output data and provides various graphic visualization options to the user.

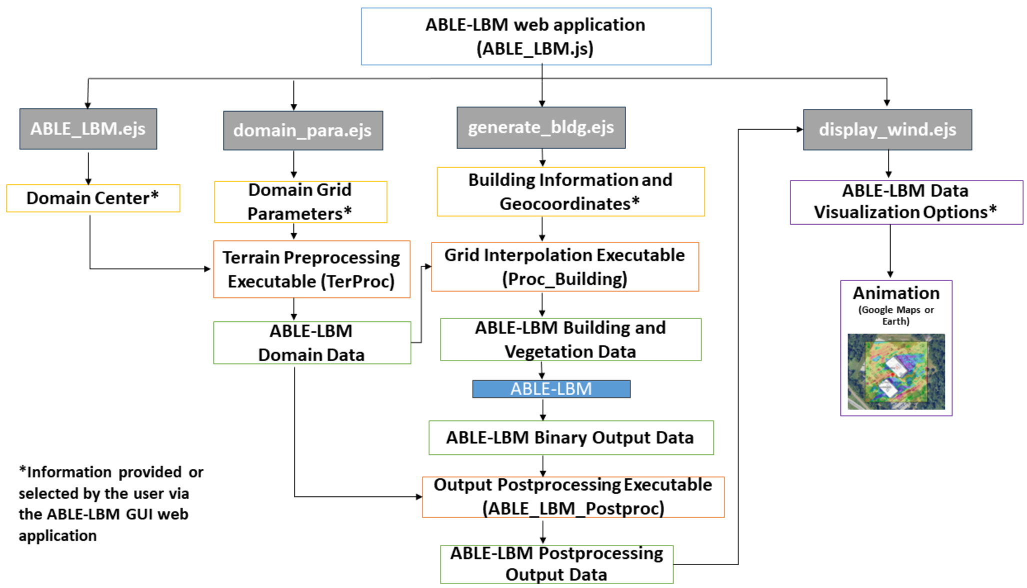

Fig. 1 ABLE-LBM web application implementation diagram

The user starts the ABLE-LBM web application on the localhost server by running the execution script "ABLE_LBM.sh" provided in the application's software package by using the command prompt. This script will automatically start the main Node.js web application "ABLE_LBM.js" and display a message such as "Server running at http://localhost:<server_port_number>/," which represents the localhost server and the server's port number currently running the ABLE-LBM web application on the user's local computer. Then, the user goes to their web browser (preferably Google Chrome) and types "http://localhost:<server_port_number>" in the web address bar to launch the web application. Note that all the input data, required executables, and output data produced and used by the application are stored in the "AppData" subdirectory of the application's software package.

3. ABLE-LBM Data Preprocessing

The ABLE-LBM terrain data processing searches and interpolates the NASA Shuttle Radar Topography Mission (SRTM)[7] terrain elevation data (WGS84) for every grid point, which covers almost the entire land area of the earth's globe from 56 south to 60 north latitudes. For this report, an example of a model run in Columbia, Maryland, was used for simple demonstration purposes. The downloaded SRTM data tiles for the ABLE-LBM web application are in the "AppData/SRTM_dat" subdirectory of the application's software package. Note that the application's software package contains the data tile that includes the Baltimore–Washington DC area only. The user can visit the United States Geological Survey (USGS) Earth Explorer[8] website to download datasets for other regions and click on their area of interest. Next, the user clicks on the "Data Sets" menu, then from the drop-down menu, selects "Digital Elevation," "SRTM," and "SRTM 1 Arc-Second Global," which is the highest-resolution SRTM data around 30 m. Then, the user clicks on the "Results" button to download the data tile. The downloaded SRTM data tile should be stored in the "AppData/SRTM_dat" subdirectory of the application's software package so the application can automatically find the correct data tile for the user's area of interest.

The ABLE-LBM web application uses Google Maps for GIS purposes, so the user can quickly identify their area of interest using images and geodetic coordinates during the ABLE-LBM preprocessing stage. As shown in Fig. 1, the ABLE-LBM EJS module "ABLE_LBM.ejs" retrieves the domain center information in latitude and longitude from the user on the GUI during the application's initial start-up through entering the latitude and longitude coordinates or by simply clicking on their area of interest (indicated by a red balloon marker on the GUI). Then, the application[2] retrieves the SRTM data of the area of interest, interpolates the terrain elevation and geocoordinates for each computation grid point, and stores the latitude and longitude values of the domain center into a local file called "ABLE_LBM_domain_center_lat_lon.txt" on the computer. Then, the user clicks on the "Set Domain (Point obs.)" button to set additional domain parameters, which calls the EJS module "domain_para.ejs," as shown in Figs. 1 and 2.

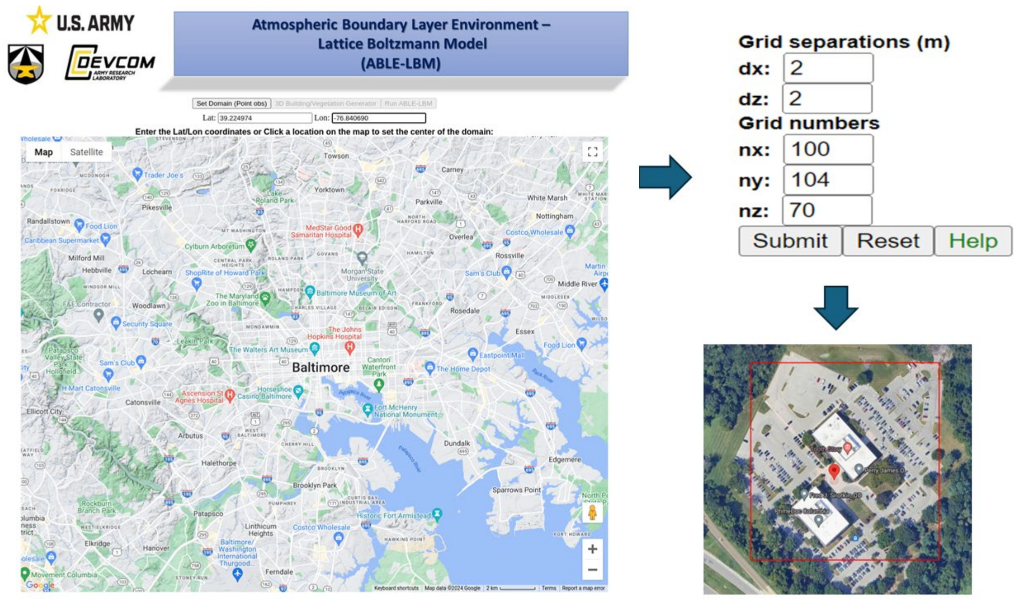

Fig. 2 ABLE-LBM set domain GUI and terrain data processing plot

The domain grid parameters (i.e., grid length in meters and grid numbers) is collected from the user on the GUI (Fig. 2), then the application stores these domain parameters into a text file called "ABLE_LBM_domain_parameter.txt." The integer "dx" parameter (in meters) is the grid spacing number in the x-direction of the 3-D ABLE-LBM domain and the integer "dz" parameter (in meters) is the level or layer spacing in the vertical direction. Note that high-resolution cases are considered when dx is less than 10 m. The integer parameters nx, ny, and nz are the number of grid points in the x-, y-, and z-coordinate directions, respectively. The application automatically adds 1 to the nx, ny, and nz grid number values to account for the ABLE_LBM's index arrays starting at 1. Then, this data is ingested into a Fortran terrain processing executable program called TerProc,[8] which creates both the gridded binary terrain elevation and domain rectangle data files. The gridded binary terrain elevation data file is called "ht_lat_long_ABLE_LBM_domain.dat," which is an unformatted binary, direct-access file that includes the elevation, latitude, and longitude of each grid, respectively, in the computational domain. The domain rectangle data file is a formatted text file called "ABLE_LBM_domain_rectangle.txt" that contains the four latitude and longitude corners of the ABLE-LBM domain, respectively. This file is used in both the data postprocessing and graphic visualization of the ABLE-LBM output on the GIS web application discussed later in Sections 4 and 5. After the terrain elevation data processing has been completed successfully, the application will display a rectangle with a red border representing the four latitude and longitude corners of the specified ABLE-LBM domain (Fig. 2). Furthermore, terrain data processing not only involves the interpolation of the terrain elevation data, but also goes through several geodetic coordinate transformations as described in Dawson.[6] The detailed description about the coordinate transformation and interpolation is described in Wang et al.[2]

Additional preprocessing information of the grid interpolation for building and vegetation data generation is also collected from the user on the GUI when the user clicks on the “3D Building Generator” button, which calls “generate_bldg.ejs” to display the ABLE-LBM Building and Vegetation Generator GUI. As shown in Fig. 3, this GUI captures information about all the buildings[2] or vegetation areas[4] in the domain within the red border through the user’s input.

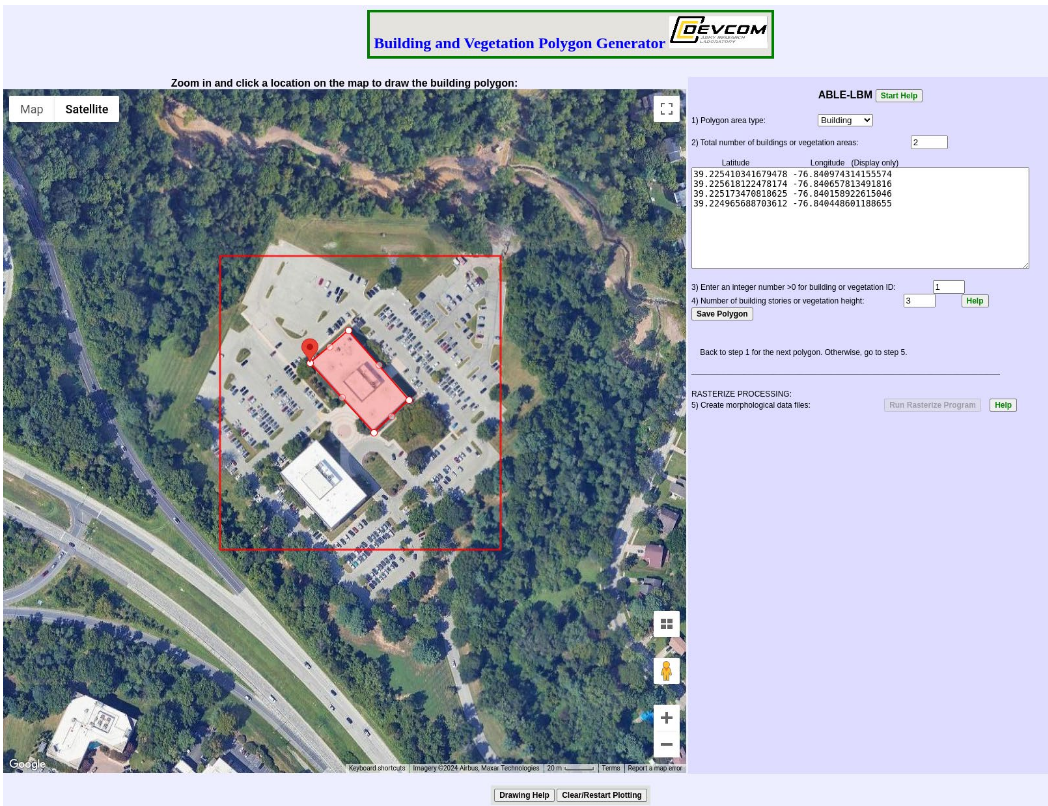

Fig. 3 ABLE-LBM building and vegetation polygon generator GUI

The building or vegetation information obtained by the user on the GUI includes the following:

- The polygon area type (building or vegetation)
- The total number of buildings or vegetation areas of interest in the domain
- For each building or vegetation area, the geocoordinates (or the latitude and longitude points) of their building corners, building or vegetation area identifier (ID) number, and number of building stories or vegetation height in meters

The geocoordinates are retrieved by the user mouse clicking on Google Maps to plot clockwise from the upper-left corner and draw polygons of each building's roof or vegetation area on the GUI. The user should zoom-in as close as possible to accurately draw the polygons for the buildings or vegetation areas. Note that the user has the option to clear and restart the drawing of the buildings by clicking on the "Clear/Restart Plotting" button on the bottom of the GUI. Once all the building or vegetation area information is entered by the user, the user should click on the "Save Polygon" button to save the drawn polygon. Then, repeat the previous steps for the next polygon until all the polygons are drawn for the buildings or vegetation areas in the user's domain. Note that during each instance of the GUI, the user should only draw polygons for all buildings or vegetation areas in the domain at a time (i.e., the user should not mismatch the drawing of polygons for buildings and vegetation areas at the same time). Once the drawing of the polygons for one building or vegetation area type is completed after running the Rasterize program, the user should click on the "3D Building/Vegetation Generator" button from the main ABLE-LBM web application and restart the Building and Vegetation Polygon Generator GUI for the other types of polygons (i.e., buildings or vegetation areas) to be drawn in the domain. For input into ABLE-LBM, the application stores the geocoordinate data of building polygons in the text file called "vertices_of_building.txt" and the geocoordinate data of vegetation areas in the text file called "vertices_of_vegetation.txt." An example geocoordinate data file, "vertices_of_building.txt," for two building polygons is as follows:

```
101 105
2
2
1
5 3
39.225410341679478 -76.840974314155574
39.225618122478174 -76.840657813491816
39.225173470818625 -76.840158922615046
39.224965688703612 -76.840448601188655
39.225410341679478 -76.840974314155574
2
5 5
39.224708038026705 -76.841215712966914
39.224915820904471 -76.840920669975276
39.224554278301149 -76.840502245368953
39.224329871610699 -76.840791923942561
39.224708038026705 -76.841215712966914
```

The first row "101 105" is the domain's number of grid points in the x- and y-coordinate directions (nx, ny), respectively, separated by white space and the second row "2" is the domain's grid length, delta_x (dx = dy = dz). All these values are obtained from the domain parameter file "ABLE_LBM_domain_parameter

.txt” as explained previously. The third row “2” is the total number of building polygons in the domain drawn by the user. The fourth row “1” is the building identifier that represents the first drawn building, which is determined by the user. The fifth row “5 3” is the total number of sides of the drawn building plus 1, and the drawn building’s total number of stories separated by white space, respectively. Therefore, in this example “5 3” represents a four-sided building with three building stories. Note that the ABLE-LBM application will always have an extra side (i.e., plus 1) in the total number of polygon sides of the drawn buildings which represents the first point of the building as required by ABLE-LBM to complete the drawn polygon. The next five rows are the geocoordinates of the building’s x- and y- grid corners in the order they were drawn on the GUI by the user. The fourth and subsequent rows repeat for each building polygon drawn in the domain. A similar process is done for drawn vegetation areas and its geocoordinate data file “vertices_of_vegetation.txt.”

After all the ABLE-LBM building or vegetation polygons in the domain have been drawn and their geocoordinate data have been stored by the application, then the user clicks on the “Run Rasterize” button on the GUI to produce rasterized gridded data for ABLE-LBM. This data is then ingested into the grid interpolation executable called “Proc_Building” for buildings, or “Proc_Vegetation” for vegetation areas. These executables are used to generate building or vegetation data in East, North, Up (ENU) coordinates, which includes the domain x and y coordinates for each building or vegetation area’s corners in the ABLE-LBM domain because the input data into ABLE-LBM must be in the model ENU coordinate system. During the building or vegetation data processing, the terrain data file “ht_lat_long_ABLE_LBM_domain.dat” previously generated during the terrain preprocessing is used for the coordinate transformation and interpolation. By using both the building or vegetation corner geocoordinates and terrain data files, the closest four grid points in the model domain (in ENU coordinates) can be found. The building or vegetation corner point’s exact x and y values (ENU coordinates in meters) is interpolated from the four nearest grid points using the standard bilinear interpolation. Each grid building or vegetation corners in the geocoordinate is searched, interpolated, and stored in a data file “building_dat.txt” for building data input or “vegetation_dat.txt” for vegetation data input into ABLE-LBM. An example building data file, “building_dat.txt,” of polygons drawn for two buildings is as follows:

```
2
1 110.40523 123.15523   5
66.092652499675751       134.78744369000196
95.133953332901001       157.71854945924133
136.31030499935150       107.91622285172343
112.63952749967575       84.702749759890139
66.092652499675751       134.78744369000196
2 115.92684 137.17684   5
43.217286666234337       56.954640157520771
69.436036666234330       79.777651727199554
105.54020333290100       39.278365488164127
82.186036666234330       14.486575676749149
43.217286666234337       56.954640157520771
```

The first row "2" is the total number of building polygons. The second row "1 110.40523 123.15523 5" contains the building identifier number determined by the user (1), calculated minimum elevation point at building base (110.40523), calculated building height in meters (123.15523), and the total number of building base points (5) separated by white space, respectively. The next five rows are the x and y coordinates in meters of the building polygon vertices, respectively, for ABLE-LBM. The second and subsequent rows repeat for each building drawn in the domain.

4. ABLE-LBM Data Postprocessing

ABLE-LBM is run manually by using the building or vegetation data as input as shown in Fig. 1 and discussed in Section 3. The output of ABLE-LBM consists of gridded binary output data that needs to be postprocessed before it can be used in the web application's graphic visualization system. This allows the model's display results of atmospheric impacts to be easily understood by the user.

The ABLE-LBM model output is a Fortran gridded binary data file called "test_uvw_fine.grd" that is opened for direct-access, unformatted input/output, with a record length of (nx+1)*(ny+1)*4 characters, where nx is the grid domain x-coordinate direction and ny is the grid domain y-coordinate direction. The ABLE-LBM binary output file consists of many time frames and has the following four variables: u, v, w, sc in each time frame, where u is the u wind component parallel to the x-coordinate axis (i.e., latitude), v is the v wind component parallel to the y-coordinate axis (i.e., longitude), w is the z wind component perpendicular to the uv plane of the map, and sc is a scalar variable. Note that scalar can be any variables (e.g., temperature, moisture, pressure, smoke) that are not related to velocity or a vector. All these values are stored into 3-D arrays of four-byte float numbers.

Since the ABLE-LBM binary output data is in Fortran array format and the ABLE-LBM web application was developed in HTML/JavaScript, wrapper code was created and developed in Java to produce an executable Java Archive (JAR) file called "ABLE_LBM_PostProc" that reads and postprocesses the ABLE-LBM gridded binary output data. This code handles the conversion of the data's array and byte order differences between Java and Fortran. Java stores data arrays with the first element's index starting at 0, while Fortran stores data arrays with the first element's index starting at 1. In addition, Java uses big endian byte order, while Fortran uses little endian byte order. The code produces the ABLE-LBM postprocessing output data by using the following preprocessing files as input: "ht_lat_long_ABLE_LBM_domain.dat" (i.e., gridded binary terrain elevation data file) and "ABLE_LBM_domain_parameter.txt" (i.e., ABLE-LBM domain parameter text file). It also uses the ABLE-LBM gridded binary output data file (i.e., "test_uvw_fine.grd") as input. The postprocessing output data that is produced from "ABLE_LBM_PostProc" consists of extensible markup language (XML) and keyhole markup language (KML) resultant files that can be used as input into the ABLE-LBM's graphic visualization system.

5. ABLE-LBM Data Visualization System

The ABLE-LBM web application displays the ABLE-LBM output data by using the EJS module "display_wind.ejs" as previously explained in Section 2. The application provides two drop-down menus for the user to choose how to view the ABLE-LBM impacts' output and transparency color levels on the GUI (Fig. 4). The ABLE-LBM impacts are displayed as color-coded on the GUI via Google Maps from colors ranging from purple to green to red/pink representing wind speeds from 0 to 13 m/s. The color bar showing these color ranges is located directly above the Google Maps area on the application.

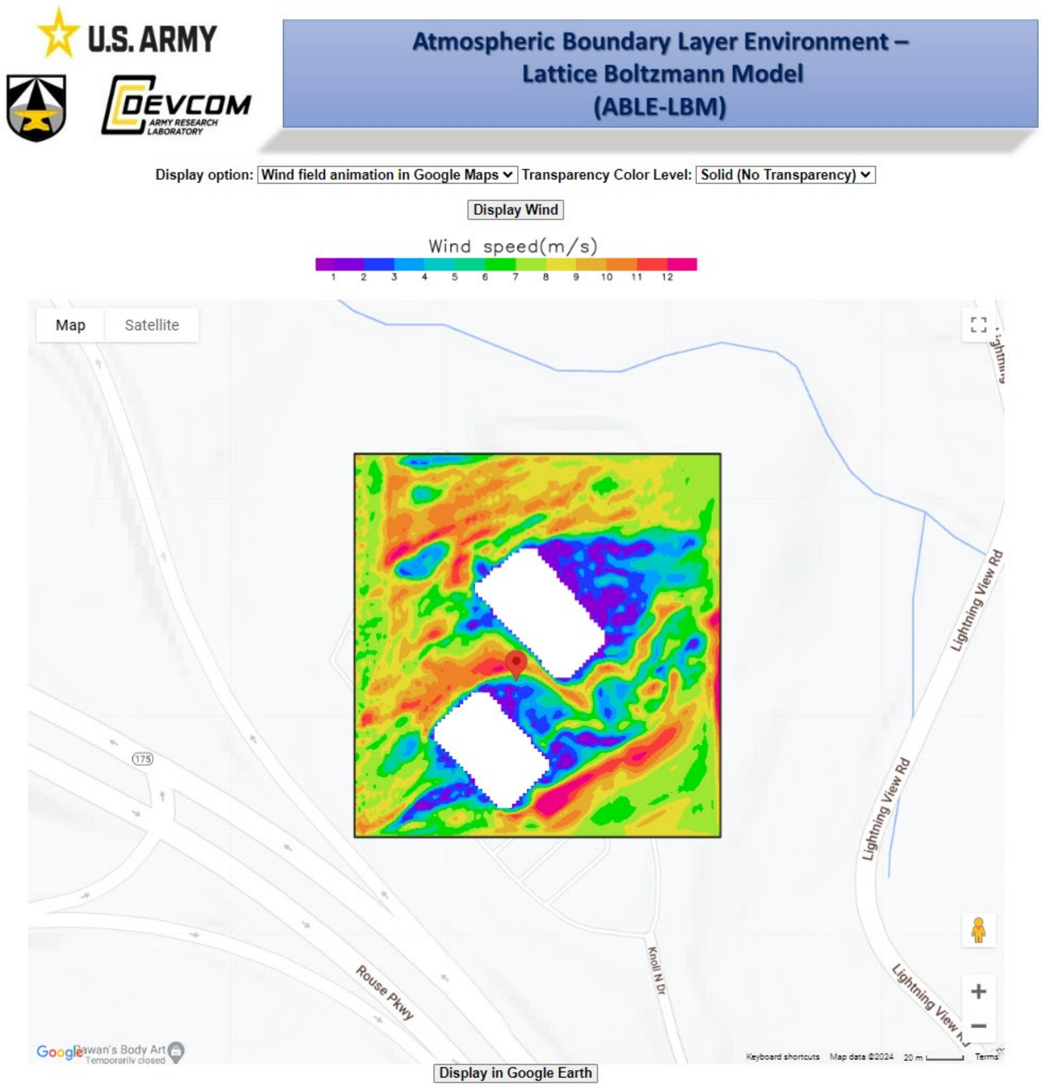

Fig. 4 ABLE-LBM display wind GUI

The 2-D display option is "Wind field animation in Google Maps." The button at the bottom of the GUI "Display in Google Earth" provides an additional display option to view the ABLE-LBM impacts in 3-D via Google Earth. The four transparency levels represent the color level of opacity or transparency for the wind field arrows, contours, or animation to be displayed in the Google Maps area on the application, which ranges from the solid (no transparency) to 75% transparency, 50% transparency, and 25% transparency. The user should click on the "Display Wind" button after selecting the display option and transparency level to view the ABLE-LBM impacts in the Google Maps area located on the application. This can be repeated as many times as needed to view the differences with changing the transparency levels on the map. By default, "Wind field animation in Google Maps" and "Solid (No transparency)" are automatically selected and displayed at the application's initial start-up of the EJS module, "display_wind.ejs."

Wind Field Animation

To enable 2-D wind field animation, the application contains a script called "rungrads," which is in the "AppData/grads" subdirectory of the application's software package. This script automatically creates a series of Portable Network Graphics (PNG) images from the ABLE-LBM's gridded binary output data by using the Grid Analysis and Display System[9] (GrADS). Then, it uses ImageMagicK[10] to edit and crop the images to be displayed on the GUI via Google Maps. The images are stored in the "AppData/public/images/animation" subdirectory of the application's software package. The application contains a technique that loops through all the images on a timer and places them on a map overlay to display them on the GUI via Google Maps. It uses the four latitude and longitude corners of the ABLE-LBM domain from the domain rectangle data file "ABLE_LBM_domain_rectangle.txt" (as discussed in Section 2) to place the map overlay over the user's domain. An example of ABLE-LBM's 2-D wind field animation in Google Maps and "25% Transparency" as the transparency color level on the GUI is shown in Fig. 5.

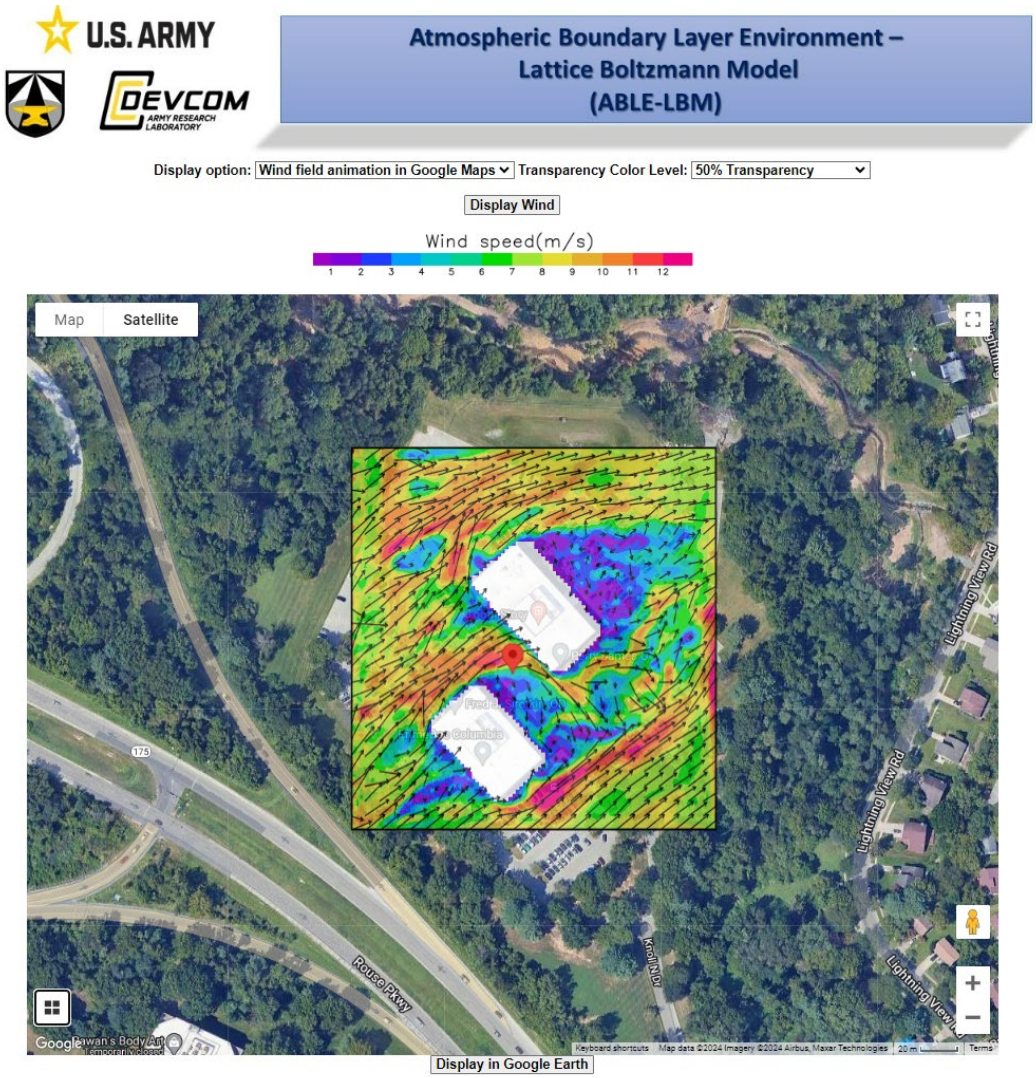

Fig. 5 **ABLE-LBM web application with 2-D wind field on Google Maps**

The "Display in Google Earth" button at the bottom of the GUI provides an additional display option to view the ABLE-LBM impacts in 3-D via Google Earth. To enable 3-D graphic visualization and animation of ABLE-LBM impacts in Google Earth, the application creates a KML file called "result_able_lbm_GE.kml" during the ABLE-LBM's data postprocessing stage. Information about the current date of execution and the location of the series of PNG images on the localhost server created by GrADS (like the 2-D wind field animation) are stored into the KML file. Also, the four latitude and longitude corners of the ABLE-LBM domain is retrieved from the domain rectangle data file, "ABLE_LBM_domain_ rectangle.txt" (as discussed in Section 2), and populated into the KML file. The altitude or distance above the ground level is set to 120 and the times for the animation in the KML file is set from 0700 to 0830 at increments of 15 min. Note that all these values are currently set temporarily and will be set to be variable based on the model's real times in the future. Like the 2-D wind field animation, the KML file is stored in the "AppData/public/images/animation" subdirectory of the application's software package. The application reads and downloads the KML file from the localhost server when the user clicks on the "Display in Google Earth" button. An example of ABLE-LBM's 3-D wind field animation in Google Earth is shown in Fig. 6.

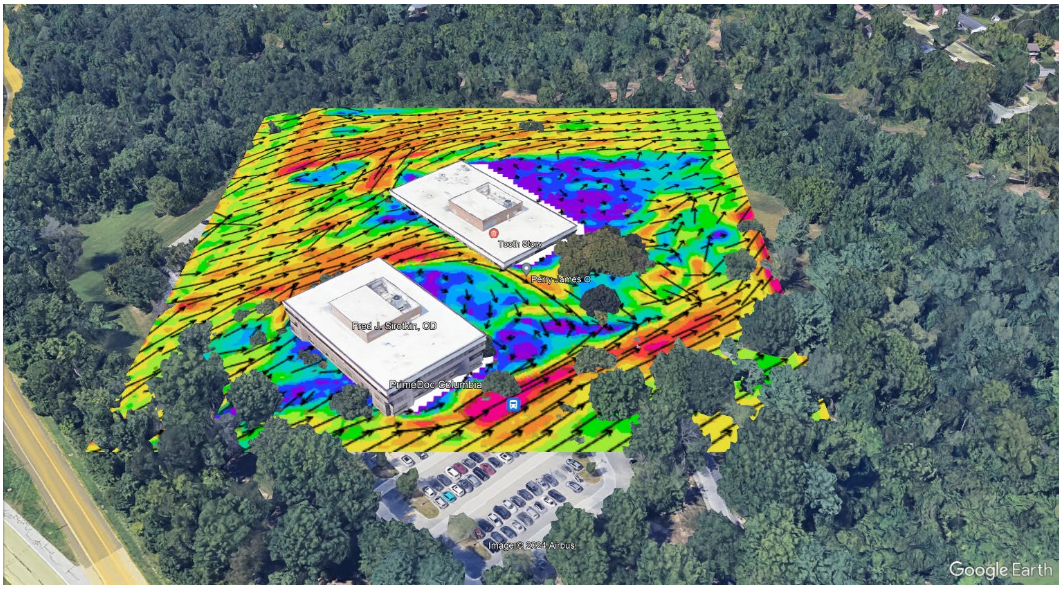

Fig. 6 ABLE-LBM web application with 3-D wind field animation on Google Earth

6. Conclusion and Future Work

ABLE-LBM developed by ARL is the next-generation atmospheric microscale model that simulates and predicts the atmospheric boundary layer's turbulent wind, temperature, and other scalar transports over complex terrain and urban areas. The ABLE-LBM web application preprocesses the required input data to the model, then after the model is run manually, it postprocesses the output data produced by the model so it can be displayed on the graphic visualization system via Google Maps or Google Earth. This allows the user to easily visualize and understand the atmospheric variable fields output from ABLE-LBM to assist in decision-making.

The future work of the ABLE-LBM web application involves the continuation of developing and improving the application including its graphic visualization system. Running ABLE-LBM is currently performed manually. Therefore, another step is to develop and integrate automated methods that handle running ABLE-LBM after the preprocessing stage and before the postprocessing stage (Fig. 1). This will allow the ABLE-LBM web application to serve as a complete automated system that performs the data preprocessing to running the model to postprocessing and the 2- or 3-D graphic visualization of the ABLE-LBM output. Additional research and development is required for the ABLE-LBM GUI web application to be fully functional on Windows computers (if this is a user requirement). Furthermore, the ABLE-LBM GUI web application stimulates and predicts turbulent wind in the atmospheric boundary layer over terrain and urban areas. Future work also includes incorporating other cases, such as temperature and other scalar transports of mass and energy in the atmospheric boundary layer. Then, atmospheric impact information can be provided to the user through the ABLE-LBM GUI web application for multiple transport types.

7. References

1. Wang Y, Decker J, Pardyjak E. Large-eddy simulations of turbulent flows around buildings using a lattice Boltzmann model. J App Meteorology. Climate. 2020;59,885–899. doi: 10.1175/JAMC-D-19-0161.1.

2. Wang Y, Huynh G, Williamson C. Integration of Google Maps/Earth with microscale meteorology models and data visualization. Comput Geosci. 2013;61:23–31.

3. Wang Y, MacCall B, Hocut C, Zeng X, Fernando HJS. Simulation of stratified flows over a ridge using a lattice Boltzmann model. Environ Fluid Mech. 2018. doi: 10.1007/s10652-018-9599-3.

4. Wang Y, Zeng X, Decker J, Dawson L. A GPU-implemented lattice Boltzmann model for large eddy simulation of turbulent flows in and around forest shelterbelts. Atmosphere. 2024;15:735. https://doi.org/10.3390/atmos15060735.

5. Node.js: an open source, cross-platform JavaScript runtime environment. OpenJS Foundation; c2024 [accessed 2024 Aug 30]. https://nodejs.org/en.

6. Dawson, L, Wang, Y. Terrain and urban data preprocessing system for the atmospheric boundary layer environment-lattice Boltzmann model. DEVCOM Army Research Laboratory (US). 2023. Report No.: ARL-MR-1068.

7. United States Geological Survey. USGS EarthExplorer. Department of the Interior (US); n.d. [accessed 2024 Aug 30]. https://earthexplorer.usgs.gov.

8. Farr T, Rosen P, Caro E, Crippen R, Duren R, Hensley S, Kobrick M, Paller M, Rodriguez E, Roth L, et al. The shuttle radar topography mission. Rev Geophys. 2007;45(2). https://doi.org/10.1029/2005RG000183.

9. Grid Analysis and Display System (GrADS). George Mason University; 2018 [accessed 2024 Aug 30]. http://cola.gmu.edu/grads/.

10. Mastering digital image alchemy. ImageMagicK Studio LLC; c1999 [accessed 2024 Aug 30]. https://imagemagick.org/.

List of Symbols, Abbreviations, and Acronyms

2-/3-D	two-/three-dimensional
ABLE-LBM	Atmospheric Boundary Layer Environment – Lattice Boltzmann Model
ARL	Army Research Laboratory
DEVCOM	US Army Combat Capabilities Development Command
EJS	Embedded JavaScript
ENU	East, North, and Up
GIS	Geographical Information System
GrADS	Grid Analysis and Display System
GUI	graphical user interface
HTML	hypertext markup language
ID	identifier
JAR	Java Archive
KML	keyhole markup language
NASA	National Aeronautics and Space Administration
PNG	Portable Network Graphics
SRTM	Shuttle Radar Topography Mission
USGS	United States Geological Survey
WGS84	World Geodetic System 84
XML	extensible markup language

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB