

# Student Simulations of Local Wildfires in a Liberal Arts Geography Course

Ted Wetherbee

Fond du Lac Tribal & Community College  
ted@fdltcc.edu

Elizabeth Jones

Fond du Lac Tribal & Community College  
ejones@fdltcc.edu

## ABSTRACT

Wildfire simulations are developed for interactive use in online geography classes under the course titled Disasters. Development of local capability to design and offer computational activities in courses at a small, rural college is a long-term activity based on integrated scientific research and education efforts.

## Keywords

Wildfires, Simulations, Computational science partnerships

## 1. INTRODUCTION

A wildfire is one of the events studied in the Fond du Lac Tribal & Community College (FDLTCC) geography course titled Disasters. This course has been offered online for several years, so spring 2020 and planned spring 2021 delivery have not been affected by COVID-19. It has minimal prerequisites and serves to satisfy a liberal education requirement for associate degrees. The material is often newsworthy enough to command headlines, but history and dramatic video footage are not interactive. We sought to add a “hands-on” component through computer simulation exercises so that students could directly explore some of these events. This is also an opportunity to add special value to an online course; we know every student has a capable computer in hand with a good connection to the Internet.

Rather than study wildfires as a general topic, we wanted students to experiment with specific wildfire variables using local and familiar geographic areas, say surrounding our campus forest or at the Big Lake meeting grounds. Simulations cannot have too fine a scale where individual trees are resolved, but at a reasonable scale, we can model fire behavior in ways interesting to students (they recognize the land) and illustrating key drivers of wildfire growth that they can control by selection: fuel, wind, topography, and moisture. Our needs for this particular class defined the modeling application.

Wildfire propagation is difficult to predict because of the complexity of fuel, terrain, induced weather, and other variables; yet modeling can still be helpful, and several kinds of models are used. WRF-fire and Sfire [5] were developed to study wildfires within the Weather Research Forecast (WRF [6]) code framework. WRF-fire and Sfire are designed for large-scale fires, say within a 50 km x 50 km area or larger, and these codes require considerable

wall-clock time to evolve as grid resolution increases. At this scale, the size of an individual cell might contain our entire campus forest. These are 2D fire-line propagation models driven by external conditions provided by WRF.

The grid size of WRF can be refined arbitrarily in horizontal directions, and the vertical resolution can also be refined by creating more pressure/eta levels. Sfire and WRF-fire run on a separate finer grid that is coupled to the finest WRF grid. It is a 2D model with inputs from the WRF grid and outputs to WRF grid variables. Sfire is used for large-scale wildfire simulations which may cover an entire mountain-side and burn for days and even weeks. Sfire scales using the same WRF mechanisms for use on multiple processes so that simulations can be run on large clusters.

The Sfire group created configuration, runtime, post-run, and display tools so that wildfire simulation results are more easily used by researchers and viewed by the public. In fact, the Sfire group offered to conduct simulations specifically for our Disasters class (given a location and date of a past fire in their catalog) so that students could examine results through the web interface.

We wanted students to set up and run simulations in some interactive fashion, and we could not run Sfire fast enough on our local machines. Our target for a real-time simulation was 15 minutes maximum. This figure can be achieved on modest workstations with a sufficiently coarse grid, and we did just this for classes in previous years using WRF-fire instead by directly modifying the stock ideal wildfire problem provided. A straight-line plume expanding in the wind direction was the consistent result, and visualization quality was poor because of the coarse grid. This is not surprising, because WRF is a meso-scale model designed to predict Earth weather.

We are more interested in small-scale fires that evolve, say, within a 1 km x 1 km area. This scale is suitable for educational uses. Our technical goals are several.

1. Problem evolution is fast enough for immediate feedback. Students—like researchers—want to see results from their experiments in short order.
2. Recognizable local terrain features are clear in visualization.
3. Students set the fire ignition point or line, fuel moisture, and wind conditions in a visual fashion.
4. Running simulations are viewable by everyone, and finished results are archived and viewable by everyone.
5. Controls for setting fire simulations, tracking jobs on the queue, and viewing results are simple and intuitive.
6. Students in 2 sections of 35 students each can make several runs over a period of 1–2 weeks.
7. An ordinary stock web browser front-end to the simulation exercise is sufficient, i.e., there are no machine-specific requirements, and no special web browser plug-ins are needed.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

## 2. HISTORY

Our problem came first. Wildfires are a local concern for practical and historical reasons. The “Fire of 1918” destroyed Cloquet, Minnesota, and much of the surrounding region, and it is still the deadliest in the United States for loss of life (453 deaths). In fall 1918, when local WWI and flu epidemic casualties were peaking, fire driven by high winds consumed towns and forests within a 1,000 mi<sup>2</sup> region in a matter of hours.



Figure 1. 1918 Fire (Duluth News Tribune, 2018).

102 years later, the region surrounding Cloquet is again mostly forest, and it is now used for a mix of production and recreation.

The FDLTCC campus was built within a planted red pine production “farm” that grew past its ideal thinning and harvesting times. It now looks much like ordinary “wild” forest in surrounding areas, except that the tallest red pines are aligned in neat rows at somewhat uniform height.

This particular stand of trees would have been harvested for telephone poles earlier had it remained in production, and there would have been periodic brushing (removing undergrowth) to reduce fire danger to the canopy. It is more of a fire hazard than necessary, and future aesthetic value of tall red pines (up to 250 feet) is eliminated by over-crowding.

In fact, there was a wildfire in the campus forest within a year after the campus was built in 1993. At that time, the undergrowth was low, so an undergrowth fire burned for several hours during the night without any damage and before notice. Tall red pines can easily survive a surface fire, and the Cloquet fire department quickly put out the fire because ordinary trucks could move between trees over low brush.

## 3. SIMULATION DETAILS

Earlier, we had developed fluid code for use at FDLTCC in collaboration with The Laboratory for Computational Science and Engineering at The University of Minnesota (LCSE). This was done in order to simulate tracer flow over complex terrain for a different problem: track flow and dispersion of a benzene cloud. This disaster occurred in Duluth-Superior during 1991 from a railroad tanker derailment and rupture off the Nemadji River Bridge.

The code tracks tracers well toward visualization of smoke and gaseous products. For fire propagation, the movement and state of the fluid can be used directly. Heat dries out vegetation, which is then easier to ignite, and hot gas rises for a chimney effect on slopes. We can add radiation from burning vegetation and then add a cellular probability function for cell ignition based on fuel, proximity to burning cells, moisture, and temperature. This allows for a physics-based simulation to a degree practical for the course.

## 3.1 Fluid Code

The fluid code Piecewise Linear Advection and Boltzmann (PLAB) is a finite volume Godunov method [4] code similar to Piecewise Parabolic Method (PPM [3]) and Piecewise Parabolic Boltzmann (PPB [11]) codes but with linear sub-cell reconstruction and representation, respectively. Linear methods are simpler and sufficient for our relatively low-speed flow fluid problems. Higher-order parabolic methods were used at this time by Woodward’s Blue Waters (BW [1]) Petascale Team to study evolution of a Sakurai’s Object-class star [10]. The star burns a different fuel (H and He) and at a vastly different scale, yet these codes are alike in key principles and implementation details that matter for successful execution.

An interesting aspect of this type of fluid code is that, while details can be complex, it is a straightforward application of conservation and ideal gas laws that students already know. We can explain the essence of how it works in visual fashion to our lower division students. This may seem to be a tall claim that a research code suitable for leading-edge astrophysical simulations on BW is something lower division students in a liberal arts course can follow in some fashion. Some explanation is due.

The Euler equations in one dimension for a compressible fluid are usually used to describe the situation.

$$\begin{aligned}\rho_t + (\rho u)_x &= 0 \\ (\rho u)_t + (\rho u^2 + p)_x &= 0 \\ E_t + (uE + up)_x &= 0\end{aligned}$$

These express the relationship of mass  $\rho$ , velocity  $u$ , momentum  $\rho u$ , pressure  $p$ , and energy  $E$  across space  $x$  and time  $t$ , and they are linked through an equation of state. They apply, but a more intuitive view of the situation is actually used.

We have cells in a 3D rectangular grid that all affect each other eventually. Yet, over a sufficiently small time increment in which sound waves move less than a cell width, we just have to figure out what crosses each face between every pair of adjoining cells in each  $x$ ,  $y$ , and  $z$  direction. Every cell has an average state: pressure  $p$ , density  $\rho$ , and velocity  $u$  in the direction across the cell face.

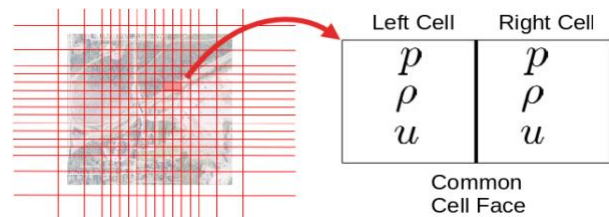


Figure 2. Cell face from a Cartesian grid.

From surrounding cells, we then reconstruct the value of each of these variables against the common cell face—left and right shown in Figure 3 for density  $\rho$ —by sub-cell distribution curves.

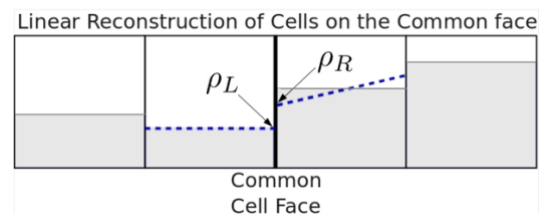
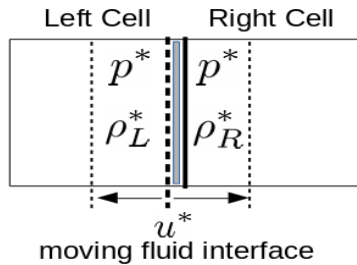


Figure 3. Reconstruction.

These curves model how real fluids behave at a sub-cell level. Our PLAB code uses linear curves sufficient for gentle fluid dynamics, while PPM uses parabolas, which are important for stronger shock waves driven by, say, thermonuclear explosions.

The Riemann solution is the state of the “star region” about the moving interface (between light dashed lines). The interface (heavy dashed line) moves with velocity  $u^*$ —left for the case shown in Figure 4. The star region expands at the speed of sound with a common pressure  $p^*$  and densities  $\rho_L^*$  and  $\rho_R^*$  on each side of the moving interface.



**Figure 4. Riemann solution.**

Mass, momentum, and energy pass from one cell to another. This is the flux. The little gray piece in Figure 4 that has moved left across the face represents flux, and we calculate this from the star region state. We do this for all cells sequentially in  $x$ ,  $y$ , and  $z$  directions, then we repeat to evolve the problem over time. For us, the solution is an illustrative video of the wildfire.

The tracer (smoke and other fire products) carried within the fluid is represented explicitly in each cell, a Boltzmann-style sub-cell distribution. This method accurately tracks tracers, and it doubles the effective resolution in each dimension toward visualization. In fact, multiple tracers can be represented within the fluid, each with its own distribution, but one tracer is enough for this wildfire simulation.

Expressing this sub-cell distribution in usable form amounts to calculating the volume fraction and moments with respect to  $x$ ,  $y$ , and  $z$  of the tracer in a  $[-1/2, 1/2]^3$  cell then limiting the slope so that our distribution function returns a value in  $[0,1]$  for each point in the cell.

$$T(x, y, z) = T_0 + T_x x + T_y y + T_z z$$

The constants  $T_x$ ,  $T_y$ , and  $T_z$  are each 12 times the moment in the  $x$ ,  $y$ , and  $z$  axis direction, respectively. Tracer is just advected with density in our wildfire code, and we set the mass  $T_0$  for the air cell at or immediately above a burning cell. After each advection step, the new tracer distribution is calculated for each cell.

Work with this function employs standard techniques from our Calculus 1 & 2 courses, a nice applied example used for these classes. It is easier to illustrate the idea first in 2D so that the density  $z$  is a function of the position  $(x,y)$  in a  $[-1/2, 1/2]^2$  square, i.e., we have a plane in space to view:

$$z = T(x, y) = T_0 + T_x x + T_y y.$$

Students can see that it really works, too: crisp, realistic tracer flow.

We feel that these methods play well to intuition and experience held by our liberal arts students. We can explain how this wildfire simulation works, more or less, depending on what students want to know.

Our PLAB code also uses the Simple Line Interface Calculation (SLIC [7]) method for solid boundaries, so we can embed the solid surface terrain, buildings, and partial solid items like tree canopy within the fluid code. Thus, dynamical features of wildfires such as eddies and chimney-like effects of slopes are simulated directly.

PLAB is not an adaptive mesh refinement (AMR) code, nor is it nestable like WRF, so we stretched the lateral and top boundaries outward from the focus of the simulation within a fine grid center. Stretching provides dampening of high frequencies generated within the fine grid, and it also provides a way to maintain outer fluid boundaries during relatively brief student simulations. Smoke does “pile up” within the boundaries in our visualizations because of these; this is a visual flaw we currently accept.

### 3.2 Visualization Code

Visualization code in various forms was built into earlier educational applications at FDLTCC. A ray casting volume rendering code Srend [9] was developed for use within BW Sakurai Object simulation code, and the same visualization code was used in new educational applications at FDLTCC, including this wildfire simulation. Running simulations wrote rendered images to website directories, then these images were drawn as available by web applications to show imagery and also turn sequences of frames into movies.

Visualization is a special challenge when the number of processes rises. Traditional strategies of dumping data to disk for post-processing do work, and BW is designed for this. BW storage can be written to by simulation XE nodes; then a separate job on XC nodes (the visualization cluster) can read the data, process it to imagery, then write results to storage. However, data can be condensed significantly through rendering in-place and deliver useful imagery during running simulations. This is a worthwhile strategy if in-core volume rendering is fast enough and if pre-defined rendering parameters are sufficient to explore simulation results.

Our BW team had investigated the idea that an in-core method for volume rendering could simplify aspects of imagery generation, handling, and delivery on BW as well as other machines toward exascale performance. Other significant benefits from embedding in-core visualization within simulation code are elimination of dependencies and delay associated with visualization codes and post-processing. This is a high-value convenience for research using petascale clusters, but it is an essential feature for our educational applications.

Of note—which may be surprising—is that scalable fluid and visualization code capable of employing every node of the largest clusters also runs perfectly on laptop and desktop machines, just slower or on smaller problems.

A rendering of stellar fusion fuel was done using 13,824 MPI ranks on BW compute nodes, and “smoke” was rendered by 32 MPI ranks on a local machine: same code, different numbers.



**Figure 5. Stellar fuel and wildfire “smoke” visualization.**

We can scale our wildfire application on machines we have available to suit the class, i.e., adjust the grid size and evolution time to get results delivered to students within acceptable limits.

### 3.3 Wildfire Simulation

We integrated the PLAB fluid code with Srend for runtime visualization, then we added variables and physics for wildfires. The tracer variable T for the simulation injected in a cell was defined to be mass of fuel (wood, grass, ...) consumed by the fire. This tracer includes smoke particles, CO<sub>2</sub>, water, and other gases.

Students define ignition points by drawing an ignition line using a mouse on a web image of a pre-defined, 1 km area surrounding the FDLTCC campus. They can switch between an image, an elevation map, and a fuel category map of the same area when they set the ignition line. The wind speed and direction can be set by clicking on a compass rose. Fuel moisture percentage is set using a slider.

Fire evolution itself is generated by a combination of physics and cellular methods. The fire grid is a 2D array at the same resolution as the fine horizontal inner grid of the fluid code. Each fire grid cell has variables:

- 1) Remaining fuel: in kg, initialized by the fuel category value by cell position.
- 2) Moisture: in percent, initialized by the student using the slider, where kg of water = kg of fuel \* percent of initial moisture.
- 3) Ignition status: burning (1) or not burning (0), initialized to 1 if on the student ignition line and 0 if not.
- 4) Energy in from burning step: in J, initialized at 0.

For each fire evolution step, there is a burning step followed by an ignition step. In the burning step for each cell with burning = 1, a fraction of remaining fuel is burned, then:

- 1) The mass burned is injected within the same fluid cell as a tracer.
- 2) The energy production (1.4 MJ/kg burned fuel) from burning is radiated. Surface cells in the line of sight can absorb energy.
- 3) Each cell transfers energy in to evaporation of moisture, and this energy is added to the fluid.

The ignition step is a cellular model but driven by physics. As the moisture content approaches zero and more touching cells are burning, the cell's probability of ignition rises. The temperature of the fluid (air) also changes the probability of ignition. Wind direction and relative elevation of cells are not used directly; the fluid code advects fluid in the direction of the wind and allows hot gas to rise. A random number is generated then input to the probability function to determine ignition.

This fire evolution model was tweaked through trials and guided by appearance of real wildfires. For limitations, we do not formally validate our models against other wildfire models or real fires but focus on educational uses; we make no claims about predicting real fires. On the other hand, we observe fire-line propagation in our models similar to that of real fires and research-grade wildfire models.

### 3.4 Delivery to Students

Our original server for this exercise was a 4-core Linux machine running an Apache webserver. There is JavaScript within the student web page so that students can navigate between views and define parameters visually, but form and CGI methods are mostly unchanged from the mid-1990s. The CGI script fire2.pl checks the

key and form variables then loads a job within the queue by writing variables to a file in the queue q2/ directory. Another CGI script queue2.pl shows the queue to students and displays results from any running simulation whenever the page is reloaded. The queue itself is processed by a perpetually running script runfire.pl that sequentially runs jobs (files in the queue directory) ordered by file creation time. The fire simulation code plab\_fire is launched by the runfire.pl script after modifying the namelist file with the run parameters, then the simulation code runs the simulation according to the namelist file.

Completed runs—with parameters, a selection of images, and a movie generated from all images—are written to the html/runs2/ directory by ID for public viewing.

### 3.5 Directions

The tutorial for the wildfire exercise comes after material and class discussion. A sequence of slides from the tutorial follow in the Appendix.

## 4. RESULTS

The wildfire activity has been used in several classes with very similar results.

### 4.1 Student Commentary

Students completed a post-simulation survey (spring 2017), and we include questions and responses from 4–7. Their comments are especially valuable; something can be done about specific complaints. We include their words complete with some minor punctuation corrections for clarity.

#### *Q4: Did you find the wildfire simulation and visualization activity engaging?*

- It was interesting. Better graphics and more variables would be nice.
- Yes, I enjoyed seeing the different effects that can cause fire to act differently.
- It was confusing and not very interesting at first, but then once I understood it better and was able to see the simulation and how my decisions affected the end result, it was much more engaging.
- Yes, I found it very interesting. I thought it was very well set up.
- Yes, after I got the simulations going and could see what they were all about, I really enjoyed it.
- I found it confusing at first, but once I got the hang of it, I did think it was engaging.
- Not really. I was kind of confused how to run it, even after directions. I did not really understand what I was trying to see or figure out. I wish it were explained to me better.
- It was kind of boring until I understood what I was doing.
- Yes, it was engaging.
- Definitely, I enjoy doing visualization activity. I get more out of doing things like that and hands on.
- Yes, but it could be a little faster and maybe updated. It did do its job, though.
- Yes, I found the simulation very engaging and enjoyable. It was very easy to navigate.
- Absolutely!
- The simulation was very neat in the fact that when you predict the way the wildfire would be, the wildfire does

something else. The fires did many different things that were unpredictable, but it showed how different each one was with the alternate factors.

- Yes, I did. It was interesting watching how things could develop though this activity.
- Yes, once I got the hang of it, it was very intriguing to see what my simulation would end up being like, being that the results are not given right away.

*Q5: What was the most effective part of this learning module?*

- How the change of different elements can affect the force of the wildfire.
- It gave me an idea of how wildfire actually works. It was nice to try new things and to be able to watch all of my runs.
- After doing this, you are able to actually visualize how different changes can affect wildfires.
- Seeing the results. Just wish it didn't take so long.
- I figured out that different spots strike up different fires. Maybe big ones or small ones.
- I would say the video were the most effective aspect of the module.
- Seeing how fast and hot fires can be and what fuel and wind factor into it.
- Actually getting to create the simulations ourselves rather than just watching ones that were already created.
- Everything about the module was effective because it shows us the dangers that could happen and gives us an opportunity on what to look out for.
- The most effective part of the module was seeing the fires counteract the predictions. The many factors that make each fire different makes them unique and spread differently. I learned that fires can spread just about anywhere, even if it's in an empty field or over a concrete highway.
- See how it all played out in the video.
- The most effective part were the different views that could be accessed by the clicking of a few simple buttons.
- That I was able to see the way the fire was going.
- Watching the fires, and how the winds and fire line play a role together.
- Having to write a summary about the simulations reinforced them.

*Q6: Were there barriers to completing the activity? If so, please discuss.*

- It was very confusing at first and lacked direction. You just kind of had to play with it and figure everything out for yourself.
- No.
- Just waiting for my simulations to run.
- I would say the only barrier would be the fact that the simulations run one at a time so it can get to be time consuming to get your simulations done. With proper planning this minor issue can be curbed.
- Yeah like sometimes it would not zoom in for me. That got super frustrating.
- I thought it was a little confusing how to find my way around the website after clicking on submit job.

- Yes, if you typed in the wrong code. You would have to re-enter everything. Also sometimes it would take a long time for the simulations to run.
- Yes, one I was sure if it was land cover in dirt over a very dry area. When trying to produce a fire, it was unsuccessful.
- Time was the major barrier. It took a lot of time from one run to another.
- I didn't encounter any issues in particular.
- Just too many people trying to complete runs at the same time.
- The barriers of completing this activity was that the video simulations did not work on my computer at home. I had to go to the school and play the video modules so I could summarize each one accurately.
- No.
- Following the directions closely. It gives you enough structure to do it.
- It was a little difficult to understand how the pass worked at first. I know it took me more than one try to really understand what I was doing even though there was a tutorial provided.

*Q7: What are your suggestions for improvement, if any?*

- Graphics and more variables.
- Maybe a better system to find your code within everyone else's code.
- I just suggest that the tutorial be supplied right away because it was very stressful trying to figure it out without it. Other than that, it was all right.
- More direction on how to use the website.
- I would try to explain the simulation better. Like inform the students what they are doing and explain to them what will happen. Make it more user friendly.
- I have no other suggestions.
- None.
- Make the process faster. This took me 5 hours to complete because of waiting time.
- Nothing really.
- Offer ways to configure computers at home like HP and Mac to play the videos from changing the settings.
- Longer video.
- I would try and maybe have weather conditions, such as light rain, rain, snow covered, just to add another seasonal variable.
- Update the system so it is less time between runs and better graphics.
- Maybe have different modules with different agriculture.
- The only thing I could suggest would be to have the simulations run faster.
- It would be interesting to pick completely different locations all around the country.

## 4.2 Notes on Student Comments

### 4.2.1 Engagement

Some student yeas for "engaging" were qualified with something like, "after I figured out what I was doing." This is not a terrible sign. A bit of initial mystery to solve is a wonderful spice. Our thoughts on the ideal instructions and user-interface are formed

about compromise between an interesting, educational puzzle and a sure-fire recipe. We definitely do shoot for pizzazz! Student opinions on the matter are there to guide us in this aspect.

#### 4.2.2 *Waiting in the Queue*

Many students complained that simulations took too much time to run. This is a familiar complaint, and this was mostly about their job waiting in the queue. Everyone appreciates speedup when it is about solving their own problem. It brings up memories of researchers complaining, say about “those chemists hogging the machine.” We agree with almost all student complaints, and this is one that can be addressed in a straightforward fashion because this is a scalable simulation code. There are practical issues to think about, however, on the idea of migrating a scalable code to a faster but remote machine.

Setting up and using a remote machine is more complex than using a local machine, and queue waiting can make optimization efforts pointless. On the other hand, it can work well. For a previous project, we used SDSC Trestles, TACC Lonestar, and LONI Queenbee successfully for remote runs on demand handled by a local intermediate portal server, and these particular machines consistently had short queue wait times for small jobs—a feature carefully sought. Currently, the XSEDE project features SDSC machines aimed squarely at this kind of on-demand service [8].

For an interesting cluster management paradigm, we also formerly used LCSE clusters for educational applications from FDLTCC. The principle for LCSE clusters was that educational usage trumps production jobs, and interactive usage trumps all. LCSE cluster users were expected to write production codes with restart dumps so that they could be killed any time then gracefully restarted. Production is important, but a 336-hour production job can be delayed a bit to accommodate educational and interactive usage.

A newer machine reduced the simulation run time from 15 minutes to 4 minutes, so we partly addressed the complaint about delay.

Longer simulation runs would be nice to do. Obviously, students wanted to see what else would have happened when their simulation ended.

#### 4.2.3 *Add Simulation Variables and Places*

Students want more variables in the simulation, namely arbitrary locations and weather conditions. We do, too. There are difficulties for some.

Real-time weather conditions are easiest to obtain because point conditions are sufficient in these fine-scale simulations. It is possible to obtain historical and contemporary forecast data from online sources then interpolate to desired position and date. A realistic method to incorporate rain or snow is an unknown to us other than by ramping up the moisture level in fuel. We might do well for this class exercise to have students simply look up current conditions or create their own. The current GUI allows wind speed and direction input as this is highly significant for wildfire evolution.

Fine-scale terrain and imagery data is obtainable online at various refinements. Fine-scale land cover data was not available at the fine scale of our simulations. We processed fine-scale imagery to 4 categories (grass, forest, buildings, and non-burnable) using Gimp raster features then edited 4-color raster images by hand over the simulation area using our knowledge of the land. Rather than try to store terrain, land cover, and imagery data for a large region (say Minnesota), or download and process data from a larger region (say continental US), we could practically prepare a selection of smaller

yet representative regions of interest and allow students to select one. These could be somewhat larger regions than our 1 km square region currently used.

#### 4.2.4 *More and Better Graphics*

These are the easiest to address. Our visualization software (Srend) can deliver several sets of imagery using different views simultaneously, and volume rendering does not take much time compared to problem evolution. We used FFmpeg to convert sequences of images to video, and we can create alternative formats that work in more browsers. MP4 for video seems to be favored today.

## 5. ACKNOWLEDGMENTS

For some things which LCSE partners did not do, writing code, developing local techniques, and defining specific FDLTCC activities were done at FDLTCC. LCSE researchers could have just done these things, surely a lot easier and faster. However, encouragement and support in the form of technical advice was otherwise available, within limits. (This activity was not funded directly.) LCSE researchers usually have answers to problems with fluid dynamics, HPC techniques, visualization, and N-body codes, and they helped make resources available for development, testing, and running educational applications. It is impossible to overstate the value of research and education partners willing to help cultivate institutional capacity.

Allocations from TeraGrid [2] and the San Diego Supercomputing Center for development and testing were available and ready to use within 24 hours of request. Similar XSEDE response (swift and certain) is available, and there are specific services available for applications of this sort, i.e., web access to applications for use in classes running on remote machines.

NASA’s project Center for Applied Atmospheric Research and Education (NASA-CAARE NNX15AQ02A) supported refinement of wildfire modeling activities, and the current FDLTCC KNL node was purchased with NASA CAARE funds. A NASA funded project at FDLTCC (Elizabeth Jones, “*Environmental Modeling And Research Experience*”, NNX11AQ96G) provided startup funding for tracer flow modeling.

FDLTCC and Minnesota State College and Universities funds sabbaticals for community college faculty with the proviso that efforts contribute to the mission of teaching and learning. Faculty research activities are viewed as important components, and internal FDLTCC support is also available. We have enjoyed strong, continuous internal administrative support for these specific efforts described.

## 6. FUTURE PLANS

The current stretched grid wildfire model still works so we are using it for the spring 2021 classes, but we are working on an AMR code as a replacement for similar simulations which involve fluid flow over complex terrain. Incorporating external data is easier with a fixed 2:1 grid refinement ratio vs. stretched grids. Also, visualization of AMR data is an Srend capability to exploit. Stretching the grid was a solution to get this problem running in time for the Disasters class. The AMR code would not evolve this wildfire problem faster using the same number of finest grid cells, but it would be easier to scale and would apply to a broader set of problems.



## 7. REFERENCES

- [1] Brett Bode, Michelle Butler, Thom Dunning, Torsten Hoefer, William Kramer, William Gropp, and Wen-mei Hwu. 2013. The Blue Waters Super-System for Super-Science. *Contemporary High Performance Computing: From Petascale toward Exascale* (1st. ed.). Chapman and Hall/CRC, Boca Raton, FL.
- [2] Charlie Catlett. 2002. The Philosophy of TeraGrid: Building an Open, Extensible, Distributed TeraScale Facility. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*, May 21–24, 2002, Berlin, Germany. IEEE Inc., Piscataway, NJ, 8. DOI: <https://doi.org/10.1109/CCGRID.2002.1017101>
- [3] Phillip Colella and Paul R. Woodward. 1984. The piecewise parabolic method for gas dynamical simulations. *J. Comput. Phys.* 54, 1 (April 1984), 174–201. DOI: [https://doi.org/10.1016/0021-9991\(84\)90143-8](https://doi.org/10.1016/0021-9991(84)90143-8)
- [4] Sergei Godunov, I. Bohachevsky. 1959. Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics. *Matematicheskii Sbornik*, 47(89), 3 (1959), 271–306. hal-01620642
- [5] J. Mandel., J. D. Beezley, and A. K. Kochanski. 2011. Coupled atmosphere-wildland fire modeling with WRF 3.3 and SFIRE. *Geosci. Model Dev.* 4, 3 (July 2011), 591–610. DOI: <https://doi.org/10.5194/gmd-4-591-2011>
- [6] J. Michalakes, S. Chen, J. Dudhia, L. Hart, J. Klemp, J. Middlecoff, and W. Skamarock. 2001. Development of a Next Generation Regional Weather Research and Forecast Model. In *Developments in Teracomputing: Proceedings of the Ninth ECMWF Workshop on the Use of High Performance Computing in Meteorology*, November 13–17, 2000, Reading, UK. World Scientific, Singapore. 269–276. DOI: [https://doi.org/10.1142/9789812799685\\_0024](https://doi.org/10.1142/9789812799685_0024)
- [7] William F. Noh and Paul Woodward. 1976. SLIC (Simple Line Interface Calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics* June 28 – July 2, 1976, Enschede, The Netherlands. Lecture Notes in Physics, Vol 59. Springer, Berlin/Heidelberg, Germany, 330–340. DOI: [https://doi.org/10.1007/3-540-08004-X\\_336](https://doi.org/10.1007/3-540-08004-X_336)
- [8] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaiher, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D. Peterson, Ralph Roskies, J. Ray Scott, and Nancy Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* 16, 5 (Sept.–Oct. 2014), 62–74. DOI: <https://doi.org/10.1109/MCSE.2014.80>
- [9] Ted Wetherbee, Elizabeth Jones, Michael Knox, Stou Sandalski, and Paul Woodward. 2015. In-core volume rendering for Cartesian grid fluid dynamics simulations. In *XSEDE '15: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, July 26–30, 2015, St. Louis, MO. ACM, New York, NY, 1–8. DOI: <https://doi.org/10.1145/2792745.2792780>
- [10] Paul R. Woodward, Falk Herwig, and Ted Wetherbee. 2018. Simulating Stellar Hydrodynamics at Extreme Scale. *Comput. Sci. Eng.* 20, 5 (Sep./Oct. 2018), 8–17. DOI: <https://doi.org/10.1109/MCSE.2018.05329811>
- [11] Paul R. Woodward, David Porter, William Dai, Tyler Fuchs, Tony Nowatzki, Michael Knox, Guy Dimonte, Falk Herwig, and Chris Fryer. 2010. The Piecewise-Parabolic Boltzmann Advection Scheme (PPB) Applied to Multifluid Hydrodynamics. In *Proceedings of the International Conference on Computational Science (ICCS 2010)*, May 31 – June 2, 2010, Amsterdam, The Netherlands. Elsevier, Amsterdam, The Netherlands, 10 pages.

## APPENDIX: SLIDES

## Wildfire Simulation and Visualization

Disasters – GEOG 2010  
Spring 2017

Slide 1

### Additional Factors

Fire growth and spread is influenced by a number of factors:

- Topography – presence of slopes, elevation
- Fuel moisture – humidity, soil moisture, presence of lush vegetation (or lack of)
- Wind conditions – wind speed, wind direction, often seasonally influenced
- Land cover – buildings and concrete versus forest or shrub

Slide 3

## The Fire Triangle

From Chapter 16 – Wildfires, we know that fires require three elements: fuel, heat, oxygen



Slide 2

## Wildfire Simulation

Let's apply these factors to the wildfire simulation interface, and look at the process of setting up a successful simulation.

Slide 4

This is what you'll see when you access the Wildfire Simulator.

You'll notice that the image shows the campus in the lower left, Interstate 35 running diagonally from the lower left to the upper right, and residential areas to the south of the campus and in the upper portion of the image.

The compass rose provides a sense of cardinal direction.

### Wildfire Simulator (background)

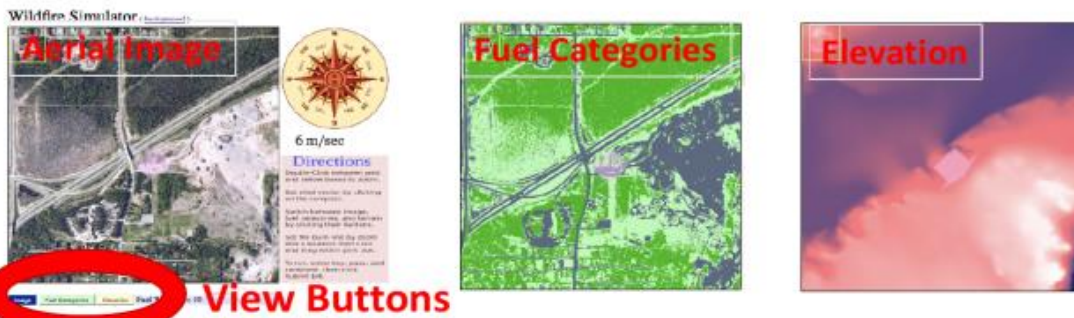


Simulation and visualization code development was funded by NASA-CAARE NNX15AQ02A.  
Elizabeth Jones (ejones@fldtcc.edu) and Ted Wetherbee (ted@fldtcc.edu)  
Fond du Lac Tribal and Community College, 2101 14th Street, Cloquet, MN 55720

Slide 5



## Three Views of the Area of Interest



1. Image: The initial view you have of the area that you'll be setting up simulation runs within is an aerial image, taken most likely from an airplane. We are most familiar with this view, and it provides us with visual reference of our area.
2. Fuel Categories: Clicking the Fuel Categories button will show you the type of fuels available. Gray and purplish are likely impervious surfaces such as concrete, and barren land such as the gravel pit across the street from the college.
3. Elevation: Clicking the Elevation button will show you a gradient surface of elevation for the area. The darker areas are lower elevation, the lighter are higher elevation. You notice the sharp contrast between the purple and pinkish colors – this represents a rapid change in elevation as can be seen when walking toward the parking lot to the north of the dorms and looking out toward Interstate 35.

Slide 6

## Setting up a Simulation Run



First steps...

1. Move your intended burn area square to the desired location by clicking in the yellow square and dragging it. To zoom in, double-click between the yellow and pink boxes. To zoom back out, double-click again between the yellow and pink boxes.
2. Click on the compass to set your wind direction and speed. The farther out from the center of the compass you click, the higher the wind speed will be. An arrow will appear in light pink on your image showing the direction and intensity of the wind.
3. Select your Fuel Water (fuel moisture) by clicking on the slider beneath the image. The default is 10%, which is a good estimate for the area shown.

*During the set-up, feel free to click around and experiment with your parameters; look at the different images available; play with the interface. It will help you become more familiar with the capabilities of the simulation.*

Slide 7

## Setting up a Simulation Run

Wildfire Simulator ( [background](#) )



24.96 m/sec

### Directions

Double-Click between pink and yellow boxes to zoom.

Set wind vector by clicking on the compass.

Switch between image, fuel categories, and terrain by clicking their buttons.

Set fire burn line by zoom into a location then click and drag within pink box.

To run, enter key, pass, and comment, then click Submit job.

First steps...

4. To create your fire burn line, which is the line from which your fire begins, you will need to be zoomed in on your intended burn area. Click on a point inside the pink box and drag to create the line, staying within the pink box.

In the image above, you see the high wind speed of 24.96 meters per second, blowing toward the east south east. What do you think these wind conditions will result in?

Slide 8

## Setting up a Simulation Run

Wildfire Simulator ( [background](#) )



24.96 m/sec

### Directions

Double-Click between pink and yellow boxes to zoom.

Set wind vector by clicking on the compass.

Switch between image, fuel categories, and terrain by clicking their buttons.

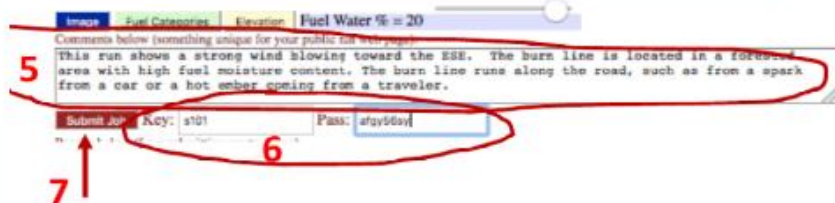
Set fire burn line by zoom into a location then click and drag within pink box.

To run, enter key, pass, and comment, then click Submit job.

First steps...

5. Enter in comments about the parameters you've selected for your run. Be specific about your conditions and your thoughts. The run will take 15-30 minutes once it is first in the queue, so you may have to come back to it later if there are other jobs waiting in line to run before yours. Your comments will help you remember what conditions you selected, and why!

6. Enter one of the Key and Pass codes that you received. The Key begins with an "s" and is followed by 3 numbers. The Pass is an 8-character code provided after each Key code. Each Key and Pass combination is unique and can only be used one time.



7. Click Submit Job!

Slide 9



## Running a Simulation Run

Once you click Submit Job, a report message will appear with code that contains the conditions you've selected. Unless this results box reports an invalid Key or Pass code, you can move on!

Submit Job Key: s630 Pass: db26hc6b

Results below (from submitting run to queue):

```
REPORT=:pass=db26hc6b:inpass=db26hc6b: time= 1493658079 qfile=/var/www/q2/1493658079_s630
water=20 key=s630 pass=db26hc6b utmx=195 utmz=89 ux=22.5 uz=10.8 sx=2 sz=21 ex=2 ez=7
phrase=This run shows a strong wind blowing toward the ESE. The burn line is located in a forested area
with high fuel moisture content. The burn line runs along the road, such as from a spark from a car or a hot
```

View Queue View Runs View Old Runs (WRFire)

8

Checking on your run (next steps)...

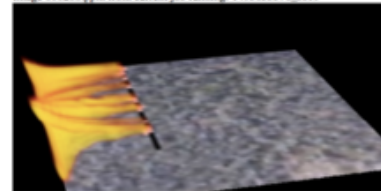
8. Once you have submitted your job, you can view where you are in the queue by clicking View Queue. All jobs that have been submitted will appear in the order by which they were submitted.

### Fire Run Queue Status

There are currently 6 jobs on the queue to run.  
Jobs from first to last on the queue (green running):

```
1: 1493653376_s619
2: 1493653666_s629
3: 1493653799_s622
4: 1493658079_s630
5: 1493658651_s330
6: 1493658875_s555
result=im=000288.ppm
```

Image 000288.ppm from current job running: 1493653376\_s619



Click your browser reload button to refresh the queue and current image.

Slide 10

## Running a Simulation Run

Submit Job Key: s630 Pass: db26hc6b

Results below (from submitting run to queue):

```
REPORT=:pass=db26hc6b:inpass=db26hc6b: time= 1493658079 qfile=/var/www
water=20 key=s630 pass=db26hc6b utmx=195 utmz=89 ux=22.5 uz=10.8 sx=2 sz=21 ex=2 ez=7
phrase=This run shows a strong wind blowing toward the ESE. The burn line is located in a forested area
with high fuel moisture content. The burn line runs along the road, such as from a spark from a car or a hot
```

View Queue View Runs View Old Runs (WRFire)

9

Checking on your run (next steps)...

9. When your run is complete, you can see it when you click View Runs. It appears as a folder in the directory. Clicking on your Key will bring you to snap shots of the run, a summary of the conditions and run time, your comments, and a link to a short video of your run.

### Index of /runs2

Name	Last modified	Size	Description
Parent Directory	-	-	-
current.jpg	2017-05-01 12:20	19K	-
s002/	2017-03-13 12:16	-	-
s003/	2017-03-13 12:48	-	-
s004/	2017-03-13 16:20	-	-
s005/	2017-03-13 16:55	-	-
s006/	2017-04-25 07:39	-	-
s007/	2017-04-26 07:08	-	-
s140/	2017-04-24 23:22	-	-
s140/	2017-04-24 22:55	-	-
s616/	2017-05-01 11:16	-	-
s619/	2017-05-01 12:21	-	-
s621/	2017-05-01 11:54	-	-

Fire run s140 : Mon Apr 24 22:55:24 2017



Wind: 12.9 m/s East, 12.6 m/s North  
Fire ignition offset from (1,15) to (1,27)  
Fuel water percent=10  
Exe:run.s140.snap.thumbnail

10. Once you have completed 5-10 runs, compose a summary as described in the simulation directions.

Slide 11