ECE 4320 Fall 2022 Course Project: Defend Ukraine

Course Instructor: Dan Molzahn, Georgia Tech ECE

1 Introduction

This project places you in charge of defending the Ukrainian power grid during the war with Russia. Your goal is to place defenses to interdict missile attacks that can be targeted at the transmission lines. Missile defenses reduce the probability that an attack targeted at a defended line will be successful. The attacker observes the locations of your defenses and then targets a set of transmission lines in order to maximize the expected amount of load shedding that will result after the system operator dispatches the generators to minimize the load shed. In other words, the project considers an intelligent attacker who plans their targets considering that 1) some of their attacks might fail, 2) they know the defender's allocation of defensive resources, and 3) the defender will redispatch the generators in response to any successful attacks. You therefore need to be more intelligent in placing defenses to mitigate the attacker's ability to cause load shedding. The remainder of this document describes the simulation and your task for the project.

2 Simulation

2.1 Dependencies and Installation

Matlab: The simulation code and all data files are given in Matlab format in `ece4320project_ukraine.zip`. The simulation code is developed for Matlab 2019a. To avoid potential incompatibilities with Matlab, you should install version 2019a or later.

Matlab Toolboxes: You will need to install Matlab's Mapping toolbox, which is available with the Georgia Tech Matlab license. To install this toolbox, click on the "Apps" tab in the Matlab command window, then click "Get More Apps". In the new window that appears, click on "Toolboxes and Products" on the "Filter by Type" list on the left side of the window. Then search for "Mapping Toolbox" and follow the instructions to install this toolbox.

Matpower: The simulation also relies on MATPOWER (`https://matpower.org`), an open-source Matlab package that is included in the `ece4320project_ukraine.zip` file. MATPOWER solves power flow and optimal power flow problems. While we will discuss MATPOWER and its associated data format in more detail during lecture, you are encouraged to install and play with this tool on your own. The MATPOWER manual provides an extensive description of the functionality of this package: `https://matpower.org/docs/MATPOWER-manual.pdf`

After installing the Mapping Toolbox, you can install rest of the entire simulation (including MATPOWER) by simply unzipping `ece4320project_ukraine.zip` and adding the associated files and folders to your Matlab path. You can do this either by using the "Set Path" button on the Matlab toolbar or by using the following command:

```
addpath(genpath('c:\path\to\unzipped\files'));
```

The simulation has a number of associated parameters that are stored in the file `setParameters.m`. These parameters can be directly edited by changing the values of the variables stored in this file.

2.2 Dataset

This project makes use of a fictitious dataset representing the Ukrainian power system with 32 buses, 10 generators, and 41 transmission lines. This dataset is available in MATPOWER format in the file `ukraine.m`. MATPOWER datasets are structured according to the format given in Section 3 and Appendix B of the MATPOWER manual.

The power system dataset is geographically placed in Ukraine, the generators approximately match those in Ukraine, and the load demands are roughly proportional to the actual load demands. However, the system is fictitious and does not match the actual power grid model. The simulation considers a single time period with a "snapshot" of the load demands.

This test case was created by Ayush Banerjee and Rachel Harris as part of an independent study project during the last several months — thank you to them for their work on this project!

2.3 Power System Model

The power system behavior is modeled using the AC power flow equations, including active power, reactive power, voltage magnitudes, and voltage angles. You can load the test case data into memory as a MatPower Case variable (mpc) with the following command: `mpc = ukraine;` The command `define_constants` will store a number of useful constants into memory that are helpful for indexing into MATPOWER test case data, as described next.

The active and reactive load demands for each bus are then stored in `mpc.bus(:,PD)` and `mpc.bus(:,QD)`, respectively. The voltage magnitude limits for each bus in per unit are stored in `mpc.bus(:,VMAX)` and `mpc.bus(:,VMIN)`. Generator data is stored in `mpc.gen`, including the bus that the generator connects to in `mpc.gen(:,GEN_BUS)`, upper and lower active power generation limits in `mpc.gen(:,PMAX)` and `mpc.gen(:,PMIN)`, and upper and lower reactive power generation limits in `mpc.gen(:,QMAX)` and `mpc.gen(:,QMIN)`. All load demand and generator limit values are given in MW and MVAr. The line parameters are stored in `mpc.branch`, including each line's terminal buses in `mpc.branch(:,F_BUS)` and `mpc.branch(:,T_BUS)`, the resistance and reactance values (in per unit) in `mpc.branch(:,BR_R)` and `mpc.branch(:,BR_X)`, and the apparent power flow limit in `mpc.branch(:,RATE_A)` in MVA. The three-phase base power for the system is stored in `mpc.baseMVA`.

An attack may result in the system being unable to supply all load demands. To model this, the loads are *dispatchable*, i.e., they are not fixed but rather can be reduced to zero if needed to make the problem feasible. The solver is incentivized to supply all loads via a large penalty in the objective function for any load shedding that might be needed.

Note that there is no coupling between the pre- and post-attack states of the system; in other words, the simulation does not consider issues such as generator ramp rates.

2.4 Defense Model

You will play the role of the defender, tasked with determining the best locations to site missile defenses. In the defense model considered in this project, each defense you place reduces the probability that an attack targeted at a particular line will be successful. This probability is specified in the value `opts.defend` in the file `setParameters.m`.

Each defense you place is associated with one transmission line. To enhance the protection, you can site multiple defenses for an individual line. Each defense operates independently such that an attack only successfully destroys a defended line if every defense fails, which occurs with probability $(1 - \texttt{opts.defend})^{(\text{num. defenses})}$.

To specify the locations of the defenses, you will set the value of a vector variable named `defense_locations`. This vector has length equal to the number of lines (41), with each entry corresponding to a line index in the Matpower data. The values in the vector `defense_locations` indicate the number of defenses sited at each location. For example, the following code would locate three defenses, two sited to protect line 1 and one sited to protect line 3.

```
defense_locations = zeros(41,1);
defense_locations(1) = 2;
defense_locations(3) = 1;
```

Additionally, even without a defense, an attack may miss a targeted line. The probability of an unsuccessful attack is dictated by the value `opts.miss` in the file `setParameters.m`.

2.5 Attack Model

The attacker chooses an set of lines to target for attack after observing the defense locations selected by the defender. The attacker seeks to maximize the expected amount of load shedding by computing, for each possible set of targeted lines, 1) the probability that each subset of targeted lines will fail and 2) the load shedding that would result if each subset of targeted lines failed, given that the generators are optimally dispatched by solving an AC optimal power flow problem that minimizes the load shedding.[1]

To model a restricted supply of missiles, the attacker is limited in the number of lines that they can target. This limit is set by the input parameter `num_attack`. The precomputed data permits choosing values for `num_attack` between 1 and 6. The simulation should be fast (less than a second) for small values of `num_attack` (less than five), but is slower when `num_attack` is greater than or equal to five.

[1] This is implemented in Matlab code which first precomputes the load shedding associated with each combination of line failures by solving many optimal power flow problems. To compute the expected amount of load shedding, the code then preforms lookups with these precomputed values to determine the load shedding and the probabilities of the outcomes that might occur from each targeted attack, given the defense locations.

2.6 Computational Tools

In order to analyze the operating conditions associated with different scenarios, you are given a function that solves the AC optimal power flow problem in the file `ece4320project_ukraine.zip`. You also have access to the load shedding that will occur from every set of line outage scenarios. As part of your work on the project, you are welcome to use, adapt, and build on this function and the outage scenarios to determine where to place the defenses.

2.6.1 AC Optimal Power Flow

The function `[sol]=runopf_with_islands(mpc,mpopts)` ("Run Optimal Power Flow With Islands"). This function solves the AC optimal power flow problem, i.e., given the loads and generator outputs, this function computes the power flows on the lines. The input `mpc` is the power system model in MATPOWER format. The second input is a MATPOWER options variable that you can use to set solver options (e.g., prevent outputs from being shown to the screen). You can read about these options in the MATPOWER manual.

To model attacked lines, you can set the lines' status to zero by modifying `mpc.branch(:,BR_STATUS)`. For example, the following code loads the Ukrainian test case data, models an attack on the third and fourth lines, and then runs an optimal power flow problem with the default MATPOWER options.

```
% Run an AC optimal power flow with two line failures

define_constants;
mpc = ukraine;
mpopts = mpoption;

mpc.branch([3 4],BR_STATUS) = 0;

[sol]=runopf_with_islands(mpc,mpopts);
```

Note that this function is identical to MATPOWER's `runopf` function, except that it can handle cases with islands, i.e., cases where the network is split into multiple disconnected components.

2.6.2 Load Shedding Data

The file `contingency_data6.mat` contains the amount of load shedding associated with each $N-k$ contingency scenario with up to $k = 6$ line failures. This data was computed by solving many optimal power flow problems and computing the corresponding amount of load shedding.

You can load this data by calling the command `load('contingency_data6.mat')`. This will load the following variables into memory:

- `samples`: A matrix storing the contingency scenarios. Each row represents a line and each column represents a contingency scenario. Entries equal to 1 denote the failure of the line corresponding to that row. For instance, a column with entries of 1 on the second and third rows would indicate a contingency scenario with the outages of the second and third lines in `mpc.branch`.

- `loadshed`: A vector with length equal to the number of contingency scenarios indicating the load shedding that occurs after the corresponding contingency scenario (after the generators are redispatched to minimize the load shedding). The load shedding is given as a proportion of the total load in the system, e.g., a value of 0.15 indicates that 15% of the total load cannot be supplied.

- `success`: A vector with length equal to the number of contingency scenarios indicating whether the optimal power flow solver successfully converged for the corresponding scenario. If the solver did not converge, the resulting load shedding is not valid.

The attacker uses this data to compute which lines to target. You may also find it useful in selecting your defense locations.

2.7 User Interface

The following function determines the attackers choice of targets for a given set of defenses:

```
[expected_load_shed, targeted_lines, outcomes] =
simulate_attack(defense_locations,num_attack)
```

The inputs to this function provide the following information:

- `defense_locations`: Specify the number of defenses to place at each line. As described in Section 2.4, you provide this information in a vector with length equal to the number of lines where each entry indicates the number of defenses to place at the corresponding line. For instance, the following code would locate three defenses, two sited to protect line 1 and one sited to protect line 3.

```
defense_locations = zeros(41,1);
defense_locations(1) = 2;
defense_locations(3) = 1;
```

- `num_attack`: Specify the number of lines that the attacker can target for their attack. This information is provided as an integer value from 1 to 6, with larger values (> 4) requiring more computing time to determine the attack.

The outputs provided by this function are:

- `expected_load_shed`: A scalar indicating the expected load shed from the attack, given the input `defense_locations` and the number of attacked lines `num_attack`. This value is given as a fraction of the total load in the system.

- `targeted_lines`: A vector indicating the lines that the attacker targets for their attack.

- `outcomes`: A structure describing the probability and load shedding corresponding to each outcome that might occur from the targeted attack. For instance, if the attacker can target four lines (`num_attack = 4`), then there are $2^4 = 16$ possible outcomes (no line failures, failure of the first targeted line, failure of the second targeted line, ..., failure of all targeted lines). This structure has three fields:

- `outcomes.failures`: A cell array with the sets of the possible line failures (successful attacks) that could occur from the attack on the targeted lines. The values in the cell array correspond to line indices (rows in `mpc.branch`).
- `outcomes.probability`: The likelihood that the corresponding failures in outcomes.failures will occur, given the `defense_locations` input.
- `outcomes.loadshed`: The load shed resulting from each possible combination of line failures in `outcomes.failures`.

A second function visualizes a selected outcome from the attack:

`displayOutcome(defense_locations,outcomes,outcomeIndex)`

The inputs to this function provide the following information:

- `defense_locations`: The number of defenses specified for each line, i.e., the same input described above.
- `outcomes`: The outcomes computed from the `simulate_attack` function, as described above.
- `outcomeIndex`: The index of the outcome that you would like to visualize, from 1 to $2^{(\texttt{num\_attack})}$.

The function `displayOutcome` is run after calling `simulate_attack` in order to visualize the results by showing a geographic map of the Ukrainian grid with the targeted lines (in red), the successfully attacked lines (dashed red), the non-targeted lines (black), the buses (black dots, and the defended lines (blue triangles, with the number of triangles indicating the number of defenses allocated to that line). The map also shows the generator outputs and upper limits on active power generation, line flows as a percentage of the flow limit, and loads with values indicating the power supplied and the power demanded. The code also outputs text to Matlab's command window that provides details on the voltage phasors, power outputs, line flows, binding constraints, etc. from the corresponding solution to the optimal power flow problem for this outcome.

The following code shows an example where two defenses are placed on lines 11 and 30 and one defense is placed on line 28, with the 16th attack outcome plotted in the figure. This outcome corresponds to the failures of every line (successful attacks on all targeted lines). The `displayOutcome` code outputs the subsequent plot below. The plot shows that all four targeted lines fail during this outcome, as all are dashed red, and the defense locations are indicated by the blue triangles. The text in the figure shows both the load shed from this outcome as well as the expected load shed across all outcomes. Note that the plots of the defense locations should be read carefully, as the blue triangles may overlap other lines. (Fixing this display issue is on my to do list!)

```
% Specify the defense locations
defense_locations = zeros(41,1);
defense_locations([11 30]) = 2;
defense_locations([28]) = 1;

num_attack = 4; % Number of lines the attacker can target
outcomeIndex = 16; % The outcome to display (successful attack on all lines)

[expected_load_shed, targeted_lines, outcomes] = simulate_attack(defense_locations,num_attack);
displayOutcome(defense_locations,outcomes,outcomeIndex)
```

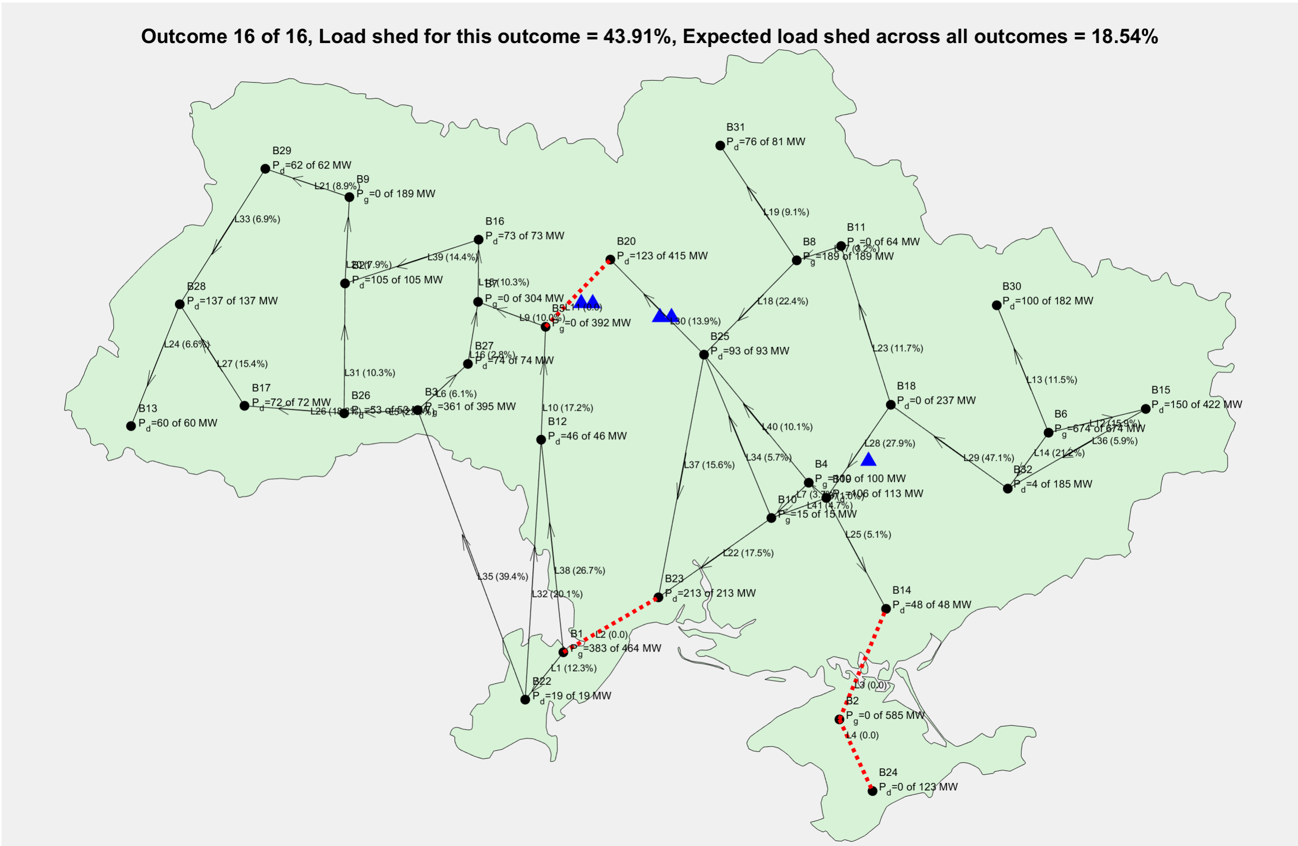

Figure 1: Example output of `displayOutcome`.

3 Project Tasks and Deliverables

You are free to choose one of the two following tasks to complete for this project. This choice will not affect your grade in the sense that high-quality work on the first task will be scored the same as high-quality work on the second task.

3.1 Task 1: Best Defense Locations

If you choose this first task, you will take the role of an advisor to the Ukrainian defense ministry responsible for optimally allocating your available defense resources to minimize the load shedding from attacks on the Ukrainian energy infrastructure. In this task, assume that you have four times the number of lines that the attacker can target available as defense resources, i.e., `sum(defense_locations)` must be no greater than `4(num_attack)`. The capabilities of your defense resources are fixed, so all values in the file `setParameters.m` cannot be changed.

Your ultimate goal is to best allocate these defense resources to minimize the expected load shedding that would result from differing attacker capabilities, i.e., differing values for `num_attack`. To advise the Ukrainian defense ministry to achieve this goal, you may want to provide answers to questions such as:

- What locations would you recommend for placing defense resources, and what is the corresponding amount of load shed that is expected to occur?

- How sensitive is the set of defense locations that you provided? In other words, are there other sets of locations that perform nearly as well at reducing the expected load shedding, or is the set of locations that you provided much better than the alternatives?

- Are there heuristics that you can provide to help allocate defense resources for other future scenarios? For instance, is it most important to defend lines in a certain part of the country or to defend lines connected to large generators?

You are welcome to use any tools that you would like to develop and analyze your recommended allocation of defense resources. For instance, you could run many simulations to do statistical analyses of the performance achieved by various approaches or to train machine learning tools.

You have two deliverables for this task:

1. A status update describing your progress on the project, due at the end of the day on December 1, 2022.

2. A written final project report, due at the end of the final exam period (December 9, 2022 at 5:30 pm eastern time). Note: there is no final exam for this course.

Your status update should be a one-page or less document describing the work you have done to date on your project. In this status update, you should also discuss the challenges or difficulties that you have encountered and your plans for addressing these challenges in order to successfully complete the project and final report by the deadline. This status update is worth 5% of your project grade.

Your final report is intended for a technical but non-specialist audience. In other words, you can assume that the reader of your report has a basic knowledge of power systems operation and therefore understands terms like "power flow", "contingencies", "load shedding", etc., but is not familiar with this simulation tool itself. You should also motivate the importance of defending energy infrastructure in Ukraine and discuss broader questions about energy infrastructure during wartime. For this purpose, your report should reference contemporary news sources and academic articles discussing these topics.

Your final report should include:

- an appropriate title for the report;

- an executive summary that briefly summarizes your results;

- an introduction discussing the context of the Ukrainian energy crisis, including references to justify and motivate the importance of your report and to provide resources for broader questions on energy infrastructure during wartime;

- a summary of the attack and defense model used in this project;

- a detailed description of the defense locations that you select for varying values of attacker capabilities (`num_attack`);

- a discussion of the methodological approach you used to select your defense locations;
- an analysis of your results with comparisons to alternative approaches that you considered;
- a description of the limitations inherent to your results, both from limitations in the attack and defense models compared to reality and from any simplifications or assumptions you make in your methodology;
- a conclusion summarizing your report.

You can use any format for this report. You should follow professional writing practices, e.g., cite the sources you use for external information (any standard citation format is acceptable, but be consistent in the citation format for all references), write clearly and concisely, organize your report in an easy to follow manner with section headings, thoroughly proofread to eliminate typos and grammatical errors, etc. You are encouraged to use clear and appropriate figures (graphs, tables, decision diagrams, etc.) in your report to illustrate your technical approach and results.

3.2 Task 2: Parameter Sensitivity Analysis

If you choose the second task, you will take the role of a technical advisor to foreign policy experts at the U.S. Department of State and the U.S. Department of Defense. Your goal is to support their efforts to ensure that Ukraine has sufficient defensive resources to protect their energy infrastructure by providing a report analyzing how much load shedding will occur if Ukraine is provided with differing numbers of defensive resources with varying capabilities. To develop this report, you will need to run multiple simulations to study how the best locations for defense resources would change as a function of their effectiveness (i.e., the value of `opt.defend` in `setParameters.m`) and the number of available resources. In other words, your report should provide a *parameter sensitivity analysis* for the load shedding with respect to both the number of defensive resources and their effectiveness.

Similar to the first task, you will need to develop a methodology for placing defensive resources to complete this second task. The difference between these tasks is that the first task aims to more comprehensively study the best places to locate a given number of defense resources, whereas the second task may less extensively analyze the placement methodology to instead focus on the impacts of varying numbers and capabilities for the defense resources.

In performing this task, you may want to consider questions such as:

- How does the ability to defend differing numbers of lines change your strategy for placing defense resources?
- What is the tradeoff curve between number of defense resources and expected load shedding? Is there a point beyond which adding more defense resources provides little benefit in reducing the load shedding?
- What is the tradeoff curve between the capabilities of defense resources and expected load shedding? Is there a point beyond which more effective defense resources provide little benefit in reducing load shedding?

- Is there a scaling that approximately indicates how many defense resources are required to respond to varying numbers of attacks? In other words, how does the number of defense resources needed to keep the expected load shedding approximately constant vary with `num_attack`?

You have two deliverables for this task:

1. A status update describing your progress on the project, due at the end of the day on December 1, 2022.

2. A written final project report, due at the end of the final exam period (December 9, 2022 at 5:30 pm eastern time). Note: there is no final exam for this course.

Your status update should be a one-page or less document describing the work you have done to date on your project. In this status update, you should also discuss the challenges or difficulties that you have encountered and your plans for addressing these challenges in order to successfully complete the project and final report by the deadline. This status update is worth 5% of your project grade.

Your final report is intended for a technical but non-specialist audience. In other words, you can assume that the reader of your report has a basic knowledge of power systems operation and therefore understands terms like "power flow", "contingencies", "load shedding", etc., but is not familiar with this simulation tool itself. You should also motivate the importance of defending energy infrastructure in Ukraine and discuss broader questions about energy infrastructure during wartime. For this purpose, your report should reference contemporary news sources and academic articles discussing these topics.

Your final report should include:

- an appropriate title for the report;

- an executive summary that briefly summarizes your results;

- an introduction discussing the context of the Ukrainian energy crisis, including references to justify and motivate the importance of your report and to provide resources for broader questions on energy infrastructure during wartime;

- a summary of the attack and defense model used in this project;

- a detailed description of your sensitivity analysis indicating how many defense resources are needed to maintain an adequate supply of power;

- a discussion of the methodological approach you used to perform your analysis;

- a description of the limitations inherent to your results, both from limitations in the attack and defense models compared to reality and from any simplifications or assumptions you make in your methodology;

- a conclusion summarizing your report.

You can use any format for this report. You should follow professional writing practices, e.g., cite the sources you use for external information (any standard citation format is acceptable, but be consistent in the citation format for all references), write clearly and concisely, organize your report in an easy to follow manner with section headings, thoroughly proofread to eliminate typos and grammatical errors, etc. You are encouraged to use clear and appropriate figures (graphs, tables, decision diagrams, etc.) in your report to illustrate your technical approach and results.

3.3 Deadlines

Regardless of the task you select for your project, your project deliverables are due by the following deadlines:

1. Project Status Report: End of the day on December 1, 2022.

2. Written Final Project Report : End of the final exam period (December 9, 2022 at 5:30 pm eastern time). There is no final exam for ECE 4320.

4 Grading Rubric

For either of the tasks, your deliverables will be graded according to the following rubric.

	Exceeds Expectations	Meets Expectations	Does Not Meet Expectations
Overall Technical Contribution (60%)			
Technical Approach (30%)	Appropriate and reasonable approach for solving the technical problem.	Overall technical approach is reasonable but has some minor flaws or shortcomings.	Approach has serious flaws or does not solve the technical problem.
Quality of Results (20%)	Capabilities and advantages/disadvantages of the technical approach are successfully and convincingly demonstrated.	The technical approach is demonstrated successfully, but has some minor inadequacies or shortcoming.	Demonstrations of the technical approach are nonexistent or severely lacking in appropriateness.
Creativity (10%)	Several aspects of the technical approach are performed with a high degree of creativity.	The technical approach has some moderately creative aspects.	The technical approach lacks creativity.
Written Report (35%)			
Structure (5%)	All paragraphs are well-organized. Section layout is logical.	Paragraphs are usually well-organized. Section layout is logical.	Paragraphs are poorly organized. Section layout is illogical.
Content (15%)	Technical discussions are clear and comprehensive. Figures enhance the readability of the report.	Technical discussions are mostly clear. Figures generally support the project results.	Technical discussions lack clarity.
Writing Quality (10%)	At most a few minor grammatical issues or word choice errors.	Sentences are generally well-written. There are a few incorrect word choices or grammatical errors that do not overly distract from the report's legibility.	Sentences are poorly written. There are numerous incorrect word choices and grammatical errors. The report's legibility is significantly flawed.
References (5%)	Detailed references and discussion explain the technical content in context of other work. Citations use a consistent format.	References adequately provide context for the project work. Citations are generally consistent.	The report fails to correctly or adequately reference related work.
Project Status Update (5%)			
Status Update (5%)		Update provided by December 1st.	