Date: 21 September 2017  
To: CMDA Capstone Teams  
From: Mark Embree  
Subject: Technical Memo 3: Brainstorming Solution Approaches

1 Executive Summary

Technical Memo 3 describes your initial attempts to formulate solution approaches for your capstone projects, and to identify algorithms that can help deliver concrete results. The goal here is to provide somewhat detailed descriptions of several different approaches to each component of your problem. Each team should post a .pdf copy of their technical memo on Canvas by 11:59pm on Thursday 28 September 2017.

2 Content of the Memo

In Technical Memo 2, you broke your capstone project into several main components, and illustrated their dependencies in a simple flowchart.

Technical Memo 3 should contain an executive summary, followed by a description of several concrete approaches that could be used to tackle each component of your capstone project.

To generate these approaches, your team will conduct a simple brainstorming exercise to think up as many potential solution strategies as possible. Many of these will be bad ideas (expensive, impractical, wrong, etc.). Do not worry about that (yet): the goal is to first generate many ideas. Only after collecting these many ideas, you will sift through them to identify those (at least two, no more than five) approaches for each component that are most promising. Describe these best ideas in your Technical Memo 3.

Perhaps you have already thought of one obvious approach to your problem. Why, then, go through this process? Through this effort, you should realize that there are often multiple ways of tackling a problem, which can yield solutions to your motivating problem that vary in accuracy, mathematical/statistical complexity, and computational practicality. The solution method you ultimately choose among these options will be dictated by the extent to which it satisfies the key criteria you identified in Technical Memo 2. (Do not worry about making such a choice now; that selection will be the subject of Technical Memo 4.)

The following two examples show representative lists of solution approaches. You should prepare a list like these for each component of your project.

Example 1: Analyzing Fluid Stability. Suppose one component of your problem is to “Identify whether a fluid, flowing in a closed cylinder and set into motion by a rotating lid, will be stable at a given viscosity.” There are a variety of strategies fluid dynamicists might take for this problem.

1. Construct a cylinder and conduct physical experiments.  
   This will require construction of the experimental apparatus, and the use of some mechanism to measure the flow (e.g., using ultrasound and tracking particles).
2. Determine an exact mathematical solution to the Navier–Stokes equations in this domain with the specified boundary conditions.

*Except for very special geometries, this is not a realistic approach.*

3. Numerically integrate the Navier–Stokes equations in time, to observe the evolution of a given steady state at given flow parameters.

*This approach, called “direct Navier–Stokes (DNS)” simulation, can be very time consuming, but gives an accurate understanding of the flow stability. In this case, the team should specify if they plan to develop a code from scratch or use black-box fluid simulators.*


*In this approach, the nonlinear Navier–Stokes equations are linearized about a steady state of the flow; one computes the rightmost eigenvalue of the linearized model. If that eigenvalue is in the right half of the complex plane, the steady state flow is unstable. This is the most common approach to this problem, but for some flows it can produce misleading results about the stability of the original nonlinear system.*

5. Model the physical flow using a lattice-Boltzmann model.

*Such models are fairly simple to implement, but are not particularly accurate.*

If writing this technical memo about the five approaches given above, a team would specify the equations involved in each approach (Navier–Stokes, linearized Navier–Stokes, etc.), or discuss specific software that could be used for each method. The goal here is to begin getting down to some level of mathematical/statistical/computational detail about the problem. Notice that *we do not yet select among these methods.*

**Example 2: Identifying Tuberculosis Bacteria.** Consider the problem of “identifying tuberculosis bacteria from an image of a slide containing a sputum sample.” There exist many nontrivial practical aspects of this problem that are required to isolate a potential bacterium from a slide image, but the core challenge is to determine whether such a candidate bacterium is tuberculosis or not. That team might have considered the following two approaches.

1. Develop a library of “eigenbacteria” from known bacteria, against which each candidate can be compared.

*This idea, adapted from the “eigenfaces” that have been proposed for facial recognition, would work (very roughly) in the following fashion. Each image in the library of known bacteria would be stacked as a column of a matrix \( A \). Compute the singular value decomposition, \( A = U \Sigma V^* \). Given an image of a potential bacterium, stack that image in a vector \( x \). The best approximation to \( x \) from the range of \( U \) is given by \( UU^T x \), so that \( (I - UU^T) x \) is the mismatch between the bacterium and the range of \( U \), which is also the range of \( A \). If this mismatch is large, the bacterium is not like the library of images, so it is judged not to be tuberculosis.*

2. Use a statistical tool called *Hu’s moments* to classify each image, with the expected values for these moments built from the library of known positive images.

*This is the approach has been used in the tuberculosis research literature.*
Each team should develop between two and five fairly independent approaches to each component of the project, as identified in Technical Memo 2.

For each approach, please include give details of the approach, any software libraries you have in mind, and the computational effort you expect this approach will require. The more detail you can provide, the better (and the easier your assessment of the methods in Technical Memo 4 will be). If you plan to scrape data, specify the tools you plan to use (e.g., the functions/packages in Python or R); if you need to optimize an objective function, specify the optimization algorithm or software.

3 Grading Rubric

This memo will be graded on a 100 point scale, according to the following rubric.

- **solution approaches:** 45 points
  - List 2 to 5 distinct approaches for each component

- **description of approaches:** 35 points
  - Describe each of the (2 to 5) identified approaches
  - Provide (basic) mathematical/statistical/computational details for each
  - Identify algorithms/software routines/packages as appropriate

- **formatting, style, grammar:** 20 points