Judge's Commentary: The Outstanding Wind and Waterspray Papers

Patrick J. Driscoll
Department of Systems Engineering
U.S. Military Academy
West Point, NY 10996
fp5543@exmail.usma.army.mil

Introduction

As so often is the case with events that iterate on an annual basis, many of the same lessons learned carry on from year to year, never losing their relevancy. Certainly, the MCM this year is no exception to the trend.

In an attempt to maintain some degree of economy in this commentary, I will resist the temptation to reiterate many of these again herein and point the interested reader to MCM commentaries previously appearing in this *Journal*.

However, there are several notable modeling issues that clearly surfaced in consideration of the Wind and Waterspray Problem that had an impact on the quality of the papers and are worth mentioning to assist teams in future competitions. In this vein, the following comments represent a compendium of observations during the final judging session and are taken in no particular order of preference or priority.

Style and Economy

As to style and clarity of the papers, it is probably sufficient to state that teams should bear in mind that they are writing to a population of modeling experts from both academia and industry who will spend a limited amount of time reading their paper. During this period, judges must assess the quality of a team's approach, the validity of their results, and the paper's completeness with regard to the modeling process. Contrast this with the hours and sometimes

The UMAP Journal 23 (3) (2002) 267–271.

days available for a professor to grade a similar project of this type, and it is apparent that teams must choose a writing style that maximizes clarity and gets across their modeling work in the most effective manner possible. Using concise and properly labeled tables and graphics to illustrate the trends and results of experimental trials that are commented on in the body of the report goes a long way towards achieving this goal.

The Specific Challenges of This Problem

The stated challenge of the Wind and Waterspray Problem was to develop an algorithm that uses data provided by an anemometer to adjust the water flow from a fountain as wind conditions change.

In a most general sense, an algorithm can be succinctly defined as a "method for the solution of all problems of a given class ... whose purpose is to describe a process in such a way that afterwards [it] can be imitated or governed by a machine" [Gellert et al. 1977, 340]. A basic characteristic of an algorithm is that it transforms given quantities (input) into other quantities (output) on the basis of a system of transformation rules. The input quantities (anemometer data) and output quantities (water flow characteristics) for the problem were clear from the problem description. The particular transformation rules for this problem were unspecified and left up to the individual teams to decide upon.

Formulating these transformation rules constituted the heart of each approach used to model the water flow and spray patterns associated with the fountain. The most predominant appeared to be Newton's Second Law of Motion, Bernoulli's formula, continuity equations, fuzzy membership sets, Poiseuille's equation, or Navier-Stokes equations, largely dependent on the assumptions that teams were willing to make.

The better papers walked the reader through the application of the approach chosen, clearly explaining exactly how each variable and parameter applied to the problem, and then used the known results of the specific approach directly.

How to Make Assumptions

Most technical report formats advise students to list and explain all their assumptions in one concise location, typically in the front portion of the report. While this advice is sound for constructing a technical report, it might be helpful to note that it is in contrast with the pattern of how assumptions occur chronologically during a modeling process. For the MCM, useful assumptions typically arise in one of two settings:

- either a team needs specific information concerning the problem that they do not have (and cannot get in the time allotted) and hence must make an assumption in order to carry on; or

- a team decides to make an assumption that simplifies some detail(s) of the problem in order to use the mathematics they are familiar with or risk not being able to complete their modeling effort in the time allotted.

Both of these situations arise naturally in the chronological flow of attacking a problem, and not during a single brainstorming effort at the onset.

When a paper contains a long list of assumptions, many of which neither get used nor justified in the modeling that follows, it is a clear indication that the team does not quite understand the roles that the assumptions play in the overall modeling effort. Such papers typically possess a very shallow or missing "Strengths and Weaknesses" section, which is supposed to constitute an analysis of one's model and results in consideration of the assumptions that were included by necessity. If a team does not know why they need a particular assumption, chances are that they will do a poor task of explaining why they made the assumption!

The lesson here is that teams should struggle mightily to make only the assumptions they need when they need them, thereby minimizing the diluting effect on model fidelity caused by an excessive number of assumptions.

The Importance of Model Validation

When all is said and done, a paper introducing a proposed algorithm must resolve the question, "Does the author provide me with sufficient evidence that it works?" While occasionally provided by way of convergence proofs, this type of evidence more commonly appears in MCM papers by way of computational testing. For the MCM, at least three categories of testing come to mind that support model *validation*:

- Once the team is convinced that their base model produces reasonable results, special cases of interest (e.g., no wind, no spread angle, etc.) should be tested.

- Recognizing that model parameters contain some amount of uncertainty, high, most likely, and low values of important parameters used in the base model should be examined by systematically altering these values and rerunning the model to see if the output results remain reasonable. For this MCM problem, these parameters might be drag coefficients, shapes of water droplets, wind speed and direction, and so on. This process essentially constitutes what is commonly referred to as *sensitivity analysis* of the parameters.

- The effects of relaxing a select number of simplifying assumptions made during the course of developing the model should be examined. However, it is fair to stress that this last category is safely performed only when time permits, because it generally requires substantial model modifications to examine the desired effects. A good example of this third category for the Wind and Waterspray problem would be adding the influence of surrounding buildings on wind speed and direction after they were previously assumed away. Such a change would be nontrivial and might consume more time than what is available.

Teams must link their computational results back to the problem that they are trying to solve. Tell the reader what to conclude from the results! This is what is referred to as *analyzing the results*. Never, ever, ever leave this task to the reader!

When the conclusions of these analyses remain the same despite changes in parameters such as those noted, it is appropriate to conclude that the model results are *robust*. These analyses also highlight any limitations of the model, which then provide a basis for recommending ways the model could be enhanced or improved in the future.

The Summary

The summary that the MCM asks for is a standalone object that should not be identical to the introduction to the paper. The summary should briefly

- state the problem,
- describe the approach taken to modeling the problem,
- state the most important results and conclusions the reader should remember, and
- mention any recommendations directly relevant to the problem.

The summary should not include a statement such as "read inside for results" or its equivalent. A good test a team can use to assess the quality of their summary is to ask, "If someone read only the summary without the rest of the report available, would it clearly tell the big picture story of what the problem was, what we did, what we concluded, and what we recommend?" As a note, most equations, code, and derivations belong somewhere else as well.

Advance Planning

With regard to time management, something that teams can do ahead of the contest is to decide

- what document-writing environment they intend to use;
- how equations will be entered and labeled;
- the outline format of the paper;

- how tables, figures, and graphics are going to look;
- how captions are going to be stated for all tables, figures, and graphics; and
- who will be responsible for what task in the final write-up.

Human nature being what it is, a sloppy or haphazard paper that looks as if it was put together 15 min before it had to be postmarked almost assuredly will be downgraded in the mind of a judge, independent of the specific results obtained.

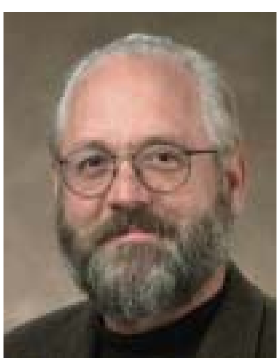
Use of Sources

Finally, the observed trend continues that teams are becoming increasingly selective with regard to the Web sites that they will trust for credible information. I also encourage teams to maintain their effort to properly document sources used to support their work. This practice explicitly recognizes the intellectual property and work of others while strengthening the quality of their paper at the same time.

Reference

Gellert, W., H. Kustner, M. Hellwich, and H. Kastner. 1977. *The VNR Concise Encyclopedia of Mathematics*. New York: Van Nostrand Reinhold.

About the Author



Pat Driscoll is Professor of Operations Research in the Department of Systems Engineering at the United States Military Academy. He holds an M.S. in both Operations Research and Engineering Economic Systems from Stanford University, and a Ph.D. in Industrial and Systems Engineering from Virginia Tech. His research focuses on mathematical programming, systems design for reliability, and information modeling. Pat is the INFORMS Head Judge for the MCM.