PSIG 1012

Controlling Complex Development Through Problem Domain Reduction: A Case Study

Jon Barley, Energy Solutions International
Greg Morrow, Energy Solutions International

This paper was prepared for presentation at the PSIG Annual Meeting held in Bonita Springs, Florida, 11 May – 14 May 2010.

ABSTRACT

“Make everything as simple as possible, but not simpler.” Who are we to argue with Einstein? In this paper we discuss a strategic approach to problem solving and its application to complex situations. The strategy is termed “problem domain reduction” and has been successfully applied to the development and testing of a simulator for modeling slack line flow for use in an on-line leak detection system.

While the concept of problem domain reduction is not new, it is a powerful strategic tool in the development of reasonably complex simulators and/or systems. The strategy allows you to focus on the essential and discard the unimportant. In doing so, insight into the underlying mechanisms is revealed whilst reducing the risk of being deceived by inconsequential artefacts.

In this paper we describe the problem domain reduction process; the selection of suitable test cases and tools to assist in analysis and verification within the reduced domain; the development of prototype simulators (which in itself is a problem domain reduction); the interplay between development and testing and finally the incremental development leading to a full domain solution.

Although the primary focus will be on the development of a simulator capable of modeling slack line flow, we will also discuss other areas where problem domain reduction has been utilized successfully to highlight the wide application of the strategy.

INTRODUCTION

Firstly it is important to understand that the strategy of Problem Domain Reduction is nothing new: it is used, without reference to the name of the strategy, in many areas of application, by many people – every day. When faced with a complex problem, one’s natural instinct is to simplify the problem. However, one has to avoid over-simplification as this will often lead to erroneous solutions to the problem. Formalizing the approach in the context of the development and analysis of a hydraulic simulator is beneficial, not least when one needs to provide a development plan with time scales and milestones. The first hurdle to overcome is often to justify the strategy.

The strategy of Problem Domain Reduction can be broken down into a few simple steps:

- **Comprehension**: Understanding the problem
- **Reduction**: Reducing the problem domain
- **Solution**: Solving the problem in the reduced domain
- **Testing**: Testing the solution in the reduced domain
- **Extension**: Extending the solution back to the full problem domain

Each of these steps may be extended to include sub-steps depending on the nature of the problem, but the core strategy remains unaltered.

JUSTIFICATION

The justification for using such an approach is that it provides clear and measurable steps on the development plan. Furthermore, it provides testing at the earliest opportunity in the development plan and therefore a key milestone. The frustration with this strategy is that success in testing in the reduced domain does not mean that the development is nearing completion – only that the most significant development has been successfully achieved. Of course, this is only a frustration for the project or product manager since typically the developer instinctively knows that extending the reduced domain solution back to the full domain, i.e. introducing complexity, may well introduce other issues. However, this can be mitigated somewhat, since we know from reducing the domain what needs to be included in extending the solution and this can be often be approached in a step-wise manner.

In the following sections we expand on the core steps in the approach and use, by way of example, the development of a slack line flow model.

UNDERSTAND THE PROBLEM

It seems obvious that understanding a problem is a prerequisite to solving the problem. However, it is essential to understand that often the quintessential essence of the problem may well be hidden in extraneous detail. Problem domain reduction relies on being able to strip away the extraneous detail and lay bare the essential fundamentals of the problem under analysis.

In terms of our example of slack line flow we have the luxury of having many sources of knowledge on the subject matter. From Nicholas[1] we know that:

- Slack line flow occurs in a liquid pipeline when the pressure falls to the vapor pressure of the fluid.
- Slack flow is essentially a separation (due to vaporization) of gas and liquid flows.
- In most pipelines, the hydraulic conditions are such that slack flow is stratified two phase flow that can be approximated well by treating the liquid component as open channel flow and neglecting the gas component altogether.
- Slack flow occurs at peaks in the pipeline elevation profile or at elevation shoulders where the elevation gradient decreases.
- Slack regions grow downstream (with respect to the fluid flow) of these elevation changes.
- Slack regions shrink from the downstream end of the slack region.
- Two continuous sections of flow separated by a slack region are effectively hydraulically disconnected, i.e. the pressure downstream of the slack region is disconnected from the pressure upstream of the slack region.
- In the slack regions the fluid velocity is much higher than in the tight sections of the pipeline.
- In the slack regions the liquid phase flowing area is smaller than the actual pipe cross-sectional area.
- Etc.

(Following Nicholas[1], the equations of flow and auxiliary equations describing slack line flow are given in appendix 1A.)

In terms of simulator development it is not unusual to have a breadth of references available detailing the mechanics and physics of a problem. Most features and enhancements that are implemented have a basis in pipeline engineering. Indeed, when trying to find the key elements in a new feature it is worth bearing in mind that many of the simplified equations and techniques employed by engineers both past and present are often themselves produced as a result of having to find good solutions to difficult questions with limited resources; they therefore inevitably arise through a process similar to problem domain reduction.

The more we understand a problem the better placed we are to reduce the problem domain to the simplest possible problem statement that still exhibits the key features. Also, the more we understand a problem the better placed we are to devise a comprehensive suite of test cases to test the proposed solution in the reduced domain.

REDUCING THE PROBLEM DOMAIN

Once the problem is understood the problem domain can then be reduced. Reducing the problem domain requires differentiating the key players (or in our case, key physics) from the incidental players and dispensing with the incidental players. Throwing away the incidental players is effectively reducing the problem domain.

It is not always straightforward to identify the incidental effects. Some effects may seem important but are not; the converse can also be true. However, if there is a clear understanding of the problem then the process of identifying the key and incidental players is easier.

There are a number of questions one can ask to ascertain the relevance of key and incidental players:

- What can I remove and still see the essential process?
- What can I simplify and still see the essential process?
- What is the simplest scenario under which the process can be observed?

Where the problem is described by a mathematical model, as in the pipe flow equations, we can look at each equation and consider if it is important or if there are any terms in the equations that can be removed or simplified.

Again, referring to our example, we know that the elevation is a key player: slack flow won't happen in a horizontal pipe (under normal conditions); similarly we know that the effect of temperature is fairly unimportant and can be discarded. Likewise, questioning and examining each of the equations, and terms in those equations, leads to the following assumptions for the reduced domain:

- Slack line flow occurs in pipes and therefore networks and pipeline equipment are secondary concerns.
- Although slack line flow can be observed in steady-state flow, the onset and growth of slack regions is transient in nature and so we cannot discard the time derivative terms in the flow equations.
- Temperature is an incidental player in slack line flow and can be neglected, i.e. the reduced domain will consider only isothermal flow.
- Elevation plays a key role in slack line flow and so cannot be discarded.

- The flowing area of the liquid is important: in the slack regions the liquid phase flowing area is smaller than the actual pipe cross-sectional area and so we cannot discard the area term or assume it to be constant.
- With the above in mind, the expansion and contraction of the pipe due to pressure and temperature plays no important role in slack line flow and we can therefore assume the pipe is rigid.
- The exact nature of the friction factor is unimportant and therefore a constant value can be used.
- The exact nature of the density correlation is unimportant: although we used a simplified bulk modulus density correlation we could just as easily assumed a constant value.
- A single fluid type can be used, i.e. there is no need for batch tracking as this is just a complication to the underlying problem.
- The exact nature of the vapor pressure equation is unimportant and therefore a constant value can be used.
- Although reverse flow can happen in a pipeline which running in slack, this is a complication rather than an essential feature of slack line flow and therefore we can consider positive flow only.

Applying this process of using problem domain reduction to the development of a slack flow model yielded the set of equations shown in appendix 1B.

As another example, if we consider implementing a compressor model in our simulator it may not be immediately obvious that you can initially dispense with the compressor map (and certainly the turbine map) and consider the compressor purely from the point of view of a device that increases the pressure head. The compressor map is just a means of calculating the pressure head increase based on the flow rate, the compressor speed, etc. Therefore our reduced problem domain is one is which there is a fixed head across the compressor.

DEVELOPING A REDUCED DOMAIN SOLUTION

The derivation of a solution in the reduced problem domain is a matter of preference and complexity. In the case of developing new functionality or features in a hydraulic simulator, one always has the option of diving straight into the existing (production) simulation engine (assuming one exists) and making the necessary adjustments to make the simulator solve the reduced domain problem. However, one must exercise caution with this approach, since most feature-rich production simulators are just that: feature-rich, i.e. they have many moving and interacting parts. Although it may well be possible to reduce the problem domain within a production simulator (for example by switching off certain equations or terms in those equations), this may have undesired side-effects that may not be immediately evident. Also, during the development of a solution one will inevitably run into problems which, due to the complexity of a fully developed production simulator, may be difficult to isolate and resolve – sometimes you just cannot “see the wood for the trees”. This approach may also have added complications in that other (unrelated) development may be occurring in the production simulator, requiring code branching etc.

An alternative approach is to develop a much simplified (prototype) simulator that is purpose-built for developing the solution to the reduced domain problem. Such simulators are used only for prototyping and are sometimes referred to as a “Lab Model” or “Toy Model”.

It may appear to be wasted effort to develop such a lab model, but it is not. The benefit of accelerated development and rapid bug resolution in an uncluttered development environment soon outweigh the costs of starting a green-field development. It is also straightforward to develop an appropriate architecture which other developers can easily navigate and contribute. Inevitably the lab model will be re-used for addressing other problems, albeit refocused and revamped, and the investment thus doubly rewarded.

To summarize the key advantages of using such a prototype:

- Rapid development: smaller code set, easier to extend
- No irrelevant detail
- Easier to find and fix problems
- No other unrelated development to contend with
- Easier to a build suitable test harness

Another benefit to developing a prototype simulator is that it gives the developer the opportunity to review the solution technique used in the production model and thus gain additional insight. Furthermore, the opportunity exists for making algorithmic improvements or improvements to the solver in a safe and easily testable environment.

TESTING IN THE REDUCED DOMAIN

Once a solution has been developed for the reduced domain solution it will need to be tested. It is essential to do as much testing as possible in the reduced domain. The reasons for this are (a) confidence and (b) problems are often easier to fix in the reduced domain solver or lab model.

It is certainly worthwhile developing at least an outline of the tests that will be required prior to developing the solution. Although the exact results of the tests may not be available, a phenomenological description of the expected outcome of the tests should be possible.

For example, in the slack flow model for the reduced problem domain we have tests that start with tight line flow and then reduce the upstream and downstream pressures in a way such that the pressure at the highest peak will fall below the vapor pressure. We know from our research into slack line flow that:

- the slack line flow will start to develop at the peak
- continuing to decrease the downstream pressure will

cause the slack region to grow
- the velocity in the slack region will be higher than the tight line flow
- the flow rate upstream of the slack region will be lower than that upstream during the growth of the slack region
- the slack region hydraulically separates the tight line flow either side of the slack region
- the mass balance of the whole system will be preserved during the process, i.e. the total mass into the system equals the total mass out of the system plus any changes in the mass inventory within the pipe.

The solution developed in the reduced domain will certainly need to be able to produce the above behavior.

The tests should be extensive and cover all anticipated operations or an abstraction thereof. Tests should therefore include the onset, growth, merge, separation, shrinkage and collapse of slack regions as well as testing under low or very flow conditions, shut-in cases, etc.

Also, one should test for extremes, since extreme cases are most likely to highlight issues. One case that initially produced “interesting” results was to run the model pipeline at a very low flow rate and allow slack flow to develop on a very shallow shoulder. Under the conditions of the test the slack flow should develop and then grow very rapidly down the shoulder. Unsurprisingly, the initial implementation of our lab model did not handle this case very well: indeed, the model went unstable. However, because we were testing the reduced problem domain in a lab model environment, we were able to quickly determine how to prevent the instability and succeed in modeling such conditions.

Another area of problem domain reduction as applied to pipeline simulation is to focus on an appropriate test pipeline that will demonstrate the efficacy of the solution under an appropriate range of conditions. For our slack line flow model, the key feature is the elevation profile (see Figure 1). This elevation profile exhibits some of the common features observed in a liquid pipeline route that crosses a mountain range, namely multiple peaks and shoulders. Varying flow rates and downstream pressures for a liquid pipeline using this elevation profile will provide a rich variety of slack line flow conditions ranging from a single slack region downstream of the first peak to multiple slack regions, including one on the shallow shoulder. Some of the these are shown in Figures 2 and 3. It is worth noting that the elevation profile we used for testing the slack flow model is actually a simplified approximation of the part of the actual elevation profile the production simulator has to cope with.

Apart from ensuring that our reduced problem domain solution provides phenomenologically correct results (which can be deemed to be the first quality gate), we would also like to have some confidence in the accuracy of our solution. This is not always straightforward: even in the reduced problem domain the problem may be difficult enough that only simulation can give comparative results. This is probably the only real downside to solving problems in a reduced domain: in the full domain we may have historical data available to us that we can use for comparison, but the simplifications made in moving to the reduced domain will mean that real data will not be as helpful in determining accuracy.

However, it may be possible to look at other ways of determining accuracy:

- Mass/Energy Balance. Demonstrating mass and energy balance in a reduced domain solution is a good first step as these are the underlying principles of macroscopic fluid flow.
- Comparison with simpler solutions. It may be possible to make further simplifications to the reduced problem domain which allow analytic, or semi-analytic, results to be generated. The reduced problem domain solver should yield the same results as the analytic or semi-analytic solver when the additional simplifications are made.
- Comparison with other simulators. If you are fortunate to be in a position where you already have a simulator that models the phenomena you are studying then direct comparisons can be made. So if you are developing a new numerical scheme to solve the pipe flow equations then comparison against an incumbent simulator can be very helpful.

For further discussions of precision and accuracy see Lagoni and Barley[3].

At this juncture it is worth noting that if you have adopted the approach of developing a Lab Model and testing the solution there, then, when implementing the solution in the production model you will often be able to mimic the reduced problem domain in the production model and compare the lab model with the production model. We found this comparison to be very useful!

During the initial development of our lab model we compared the results of running simulations until a steady-state had been achieved and then comparing the results with a steady-state solution developed in a spreadsheet.

It is at this point that, if the main thrust of solving the problem in the reduced domain has been undertaken in a lab model, one can look at implementing the solution in the production model. This approach allows the result from the lab model and production model to be compared in order to ensure that no bugs have crept into either model.

EXTENDING THE PROBLEM DOMAIN

In getting to the reduced problem domain, certain simplifications are made; when extending back to the full domain these simplifications need to be removed and the complexities and incidental effects put back into the model.

As mentioned earlier, extending the problem domain can be undertaken in a step-wise manner. It is deemed good development practice to undertake the high risk development early on in the development lifecycle. This is also good practice when extending the problem domain in a stepwise manner: include the most significant and high risk effects first. In terms of the slack flow modeling problem there are two obvious candidates for the title of highest risk tasks: including the thermal model and handling reverse flow.

If a prototype lab model has been built and tested it is natural to want to add some, if not all, of the missing features in this environment. However, one must guard against doing too much in the lab model since it will provide only diminishing returns. Certainly some of the higher risk steps can be undertaken in a lab model, and this is good use of a lab model, but the closer in complexity it gets to the production model, the less benefit there is in maintaining and developing in two environments. Once a solution to the reduced domain problem has been developed and tested in the lab model, the lab model should be used judiciously. Another consideration here is that the architecture of a lab model may differ significantly from the production model: any solution developed in the lab model will have to be later fitted into the production model, so it is probably wise to avoid adding too much complexity into the lab model and to switch development to the production model.

Implementing a successful lab model prototype in a real world application raises new complications. Once one leaves the sterile laboratory environment, one is confronted with the dirt and grime of the real world, at least metaphorically. A prototype automobile that is only ever driven on a test track does not need a trunk or rear-view mirrors, but an automobile that is expected to drive on city streets must have these features.

Real world pipeline configurations are much more complex than those required in a laboratory flow model. You can limit a laboratory model to a single continuous pipe, with the optional capability to set a flow boundary condition to zero to simulate a closed valve at one end. A real world configuration has to accommodate block valves, check valves, pumps, and other pipeline equipment. Furthermore, a real world configuration has to model much more than straight pipe; it must accommodate junctions, laterals, looped lines, and reversible pipes.

The configuration also gains additional complexity through dynamic connectivity changes. In any pipeline more complex than A to B, valve positions determine which segments are connected, shut-in, or reversed, and which injections and deliveries are active. A pressure instrument providing a pressure boundary condition can be hidden by a closed valve, requiring a zero flow boundary condition in its place.
Thus, a real world model segment can consist of multiple pipe segments joined end-to-end or separated by check valves, instead of a single straight pipe. Which pipe segments comprise a model segment can change between model steps, putting constraints on how state flags are stored and interpreted. A pipe segment can be added to a model segment in reverse, right-to-left instead of left-to-right.

In a laboratory prototype, one can build the matrix of hydraulic equations by walking a single straight pipe left to right. When building its matrix, the real world application has a greater infrastructural requirement. Additional hydraulic equations are written to join pipe segments. Check valves contribute to different hydraulic equations depending on local hydraulic conditions. This further complicates the model control structures, which must detect and re-run the model step if hydraulic changes cause a check valve to change state.

A prototype model also typically assumes a single fluid with simple properties. A real world application must be able to model multiple fluids with changing properties, moving through the model segment, whether it is propane adjacent to ethylene, gasoline next to diesel, or a crude oil whose API gravity slowly drifts as the mix from different suppliers changes. The simple single fluid continuity equation must be replaced with more complex sums that correctly account for the total mass and the pressure and temperature dependencies when one or more fluids with different properties are within the model calculation interval over which the continuity equation is written.

Real world applications, particularly real-time applications, have stringent uptime and availability requirements. In the laboratory, a prototype can crash or give nonsensical results without consequence, except insofar as it gives the developer something new and interesting to investigate. In the real world, a model that crashes or gives nonsensical results causes a major issue. Depending on the pipeline operating company and its regulatory requirements, the unavailability of a real-time application like leak detection may be considered a safety violation that requires the pipeline to be shut down.

Hence, a real-world application must be more robust against failure than a laboratory prototype, even while the other requirements of a real world model compared to a laboratory prototype are making it more complex and thus potentially more subject to unanticipated failure modes. This robustness requirement extends to initialization. A real-world application must be able to reliably start under virtually any conditions in order to meet its availability requirements.

The considerations above explain the potential lag between solving the problem in the reduced domain and solving the problem in the production simulator. Because of the additional features in the production simulator many more tests need to be devised. Ultimately the production simulator will need to be fed with real world data (assuming it is available) and soak testing using many months of real data will provide the ultimate test.

OTHER EXAMPLES

There are many other examples where problem domain reduction can be, or has been, used successfully. The most obvious and perhaps most pertinent example is that of construction a pipeline model configuration.

In terms of a pipeline model, reducing the problem domain requires identifying which configuration elements can be omitted, or their configuration made as simple as possible, and which model options/parameters should be selected to make the model as simple as possible whilst still capturing the major features. The reduced domain configuration will be one which models the key features with the minimum of complexity. It may not be hugely accurate but will be as a basis from which a more accurate model can be developed.

Extending the model back to the “full” domain is a process of adding in detail. However, there is obviously a level beyond which adding additional detail provides no further benefit in terms of accuracy and only adds to the model complexity. This subject is covered somewhat by Bachman and Goodreau[2].

CONCLUSIONS

The strategy of reducing the problem domain was instrumental to the success of developing a robust and stable slack line flow model.

Reducing the problem domain is a good strategic approach for analysis and implementation of complex problems. It provides a clear path for development and also provides key milestones

The process of reducing the problem domain requires proper analysis of the problem and therefore insight. If applied appropriately, the highest risk elements of a development can be addressed early on in the development lifecycle. The use of prototype (lab) models is encouraged as this provides an unencumbered and uncluttered development environment and thus enables rapid analysis of alternative solutions.

Testing is a key part of the strategy. Early testing in the reduced domain not only demonstrates progress, but also starts to build confidence in the solution approach. Developing a comprehensive set of tests prior to development of a solution provides further insight into the problem and also ensures that the solution is not limited by lack of consideration of specific phenomena.

Extending the reduced problem domain back to the “real world” problem domain should be done in a considered and, perhaps, stepwise manner. The artifacts with the highest perceived risk should be addressed first. Keep in mind that a lab model prototype is a long way away from a fully developed production simulator.

AUTHORS

Jon Barley is Director of Simulation and Modeling at Energy Solutions International, Ltd. in Stockton-on-Tees, UK. Jon has over 25 years experience in Computational Fluid Dynamics, 17 of which have been in the pipeline industry. He currently leads and manages the development of Energy Solutions’ pipeline modeling software. Jon has a Bachelors degree in mathematics from the University of York and a Doctorate in Computational Fluid Dynamics from the University of Reading.

Greg Morrow is a Lead Developer at Energy Solutions, Inc. in Houston, Texas. He has 13 years experience in pipeline simulation and specializes in real-time modeling and leak detection. He is currently the lead developer for the PLDS hydraulic model within Energy Solutions’ PipelineManager®. Greg has a Bachelor’s degree from the University of Notre Dame and a Master’s degree and a Ph.D. in physics from Rice University.

REFERENCES

1. “Simulation of Slack Line Flow: A Tutorial”, R. Edward Nicholas, PSIG, 1995

2. “Less is More: Accuracy versus Precision in Modeling”, Susan Bachman and Mary Goodreau, PSIG, 2000

3. “On Simulation Accuracy”, Per Lagoni and Jon Barley, PSIG, 2007

ACKNOWLEDGEMENTS

The authors wish to thank Energy Solutions International for their kind permission to present and publish this paper.
We specifically wish to thank Pradeep Ittycheria and Hugh Robinson for their support, encouragement and reviews and especially Laura Holder who, as a project manager, took the bold step of embracing this strategy on a critical project.

APPENDIX 1A. FULL DOMAIN FLOW EQUATIONS

Mass Balance

$$\frac{\partial}{\partial t}(A\rho)+\frac{\partial}{\partial x}(Av\rho)=0$$

Momentum Balance

$$\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial x}+\frac{1}{\rho}\frac{\partial P}{\partial x}+g\frac{\partial h}{\partial x}+\frac{fv|v|}{8R_h}=0$$

Thermal Balance

$$\rho c_v\left(\frac{\partial T}{\partial t}+v\frac{\partial T}{\partial x}\right)=-T\left(\frac{\partial P}{\partial T}\right)\bigg|_{\rho}\frac{\partial v}{\partial x}+\rho\frac{fv^2|v|}{8R_h}-\frac{4U_w}{D_0}(T-T_g)$$

Area Equation

In tight regions:

$$A=A_0\left(1+\frac{D_0}{\tau E}(P-P_0)+\eta(T-T_0)\right)$$

In slack regions we select a flowing area such that vapor pressure is maintained.

Hydraulic Radius

$$R_h=\frac{D}{4}\left(1-\frac{\sin(2\theta)}{2\theta}\right)$$

θ is defined as the angle, relative to the center of the pipe, from the bottom of the pipe to where the top of the fluid intersects the pipe.

Friction Factor

$$\frac{1}{\sqrt{f}}=-2\log\left(\frac{2.51}{\mathrm{Re}\sqrt{f}}+\frac{k}{3.7D}\right)$$

Density Correlation

The fluid is described by the API fluid correlation:

$$\rho=\rho_0V_{Pc}V_{Tc}$$

where V_{Pc} is the volume correction factor for pressure and V_{Tc} is the volume correction for temperature. V_{Tc} is given in look-up tables that have been derived from API data.

V_{Pc} is given by the formula

$$V_{Pc}=\frac{1}{1-F(\gamma,T)(P-P_A)}$$

$$F(\gamma,T)=0.00001e^{K(\gamma,T)}$$

where

$$K(\gamma,T)=-1.99470+0.00013427T+(0.79392+0.0023260T)\gamma^2$$

and

$$\gamma=\frac{\rho_0}{\rho_w}=\frac{141.5}{API+131.5}$$

Appendix 1B. Reduced Domain Flow Equations

Mass Balance

$$\frac{\partial}{\partial t}(A\rho)+\frac{\partial}{\partial x}(Av\rho)=0$$

Momentum Balance

$$\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial x}+\frac{1}{\rho}\frac{\partial P}{\partial x}+g\frac{\partial h}{\partial x}+\frac{fv^2}{8R_h}=0$$

Thermal Balance

The system is isothermal.

Area Equation

In tight regions the area is constant.
In slack regions we select a flowing area such that vapor pressure is maintained.

Hydraulic Radius

$$R_h=\frac{D}{4}\left(1-\frac{\sin(2\theta)}{2\theta}\right)$$

θ is defined as the angle, relative to the center of the pipe, from the bottom of the pipe to where the top of the fluid intersects the pipe.

Friction Factor

A constant friction factor is used

Density Correlation

Density is a constant:

$$\rho=\rho_0$$

Or alternatively a simplified Bulk Modulus correlation:

$$\rho=\rho_0\left(1+\frac{P}{BM}\right)$$

APPENDIX 2. REDUCED DOMAIN TEST CONFIGURATION AND SOME SOLUTIONS

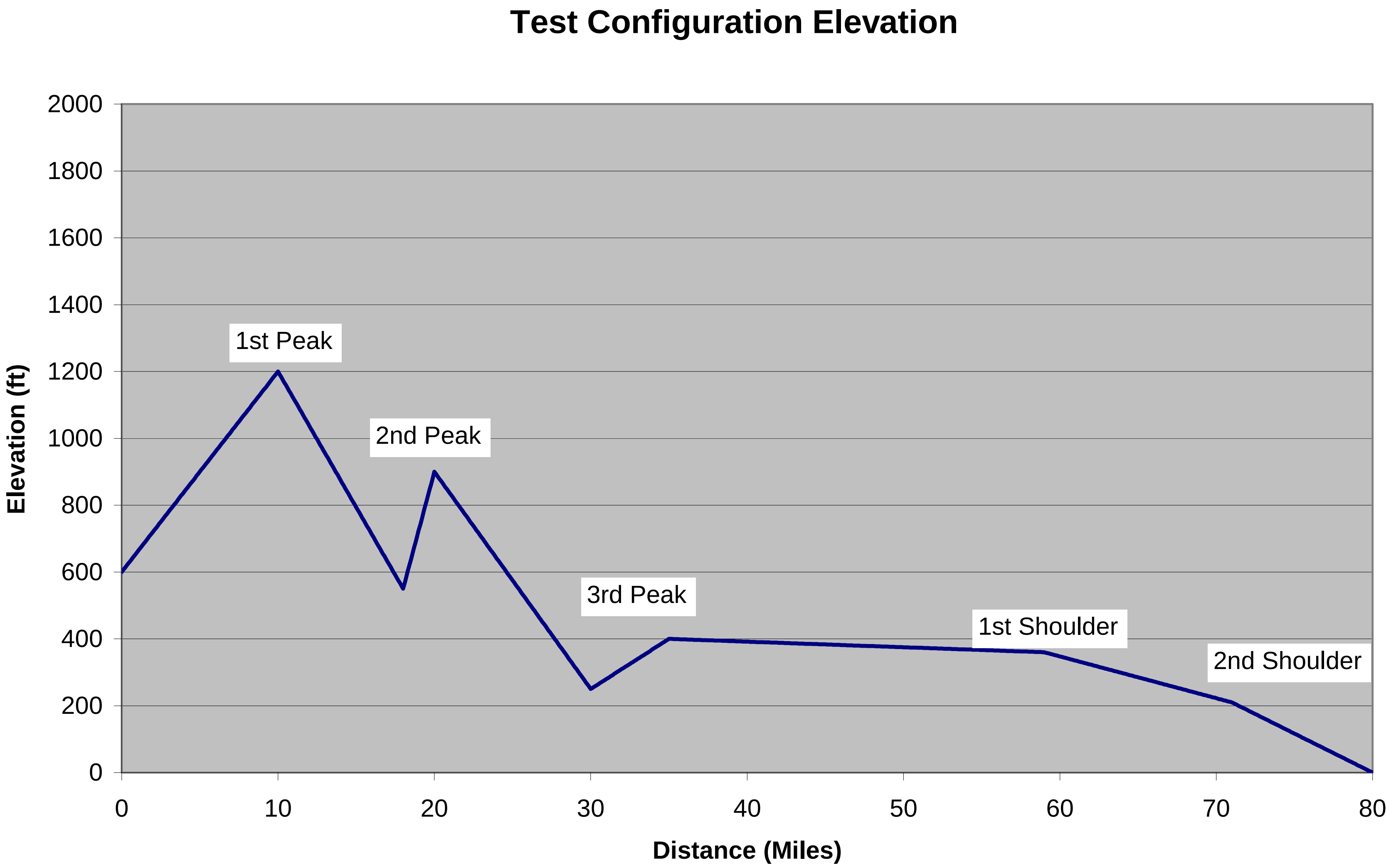

Figure 1 - Test Configuration Elevation Profile

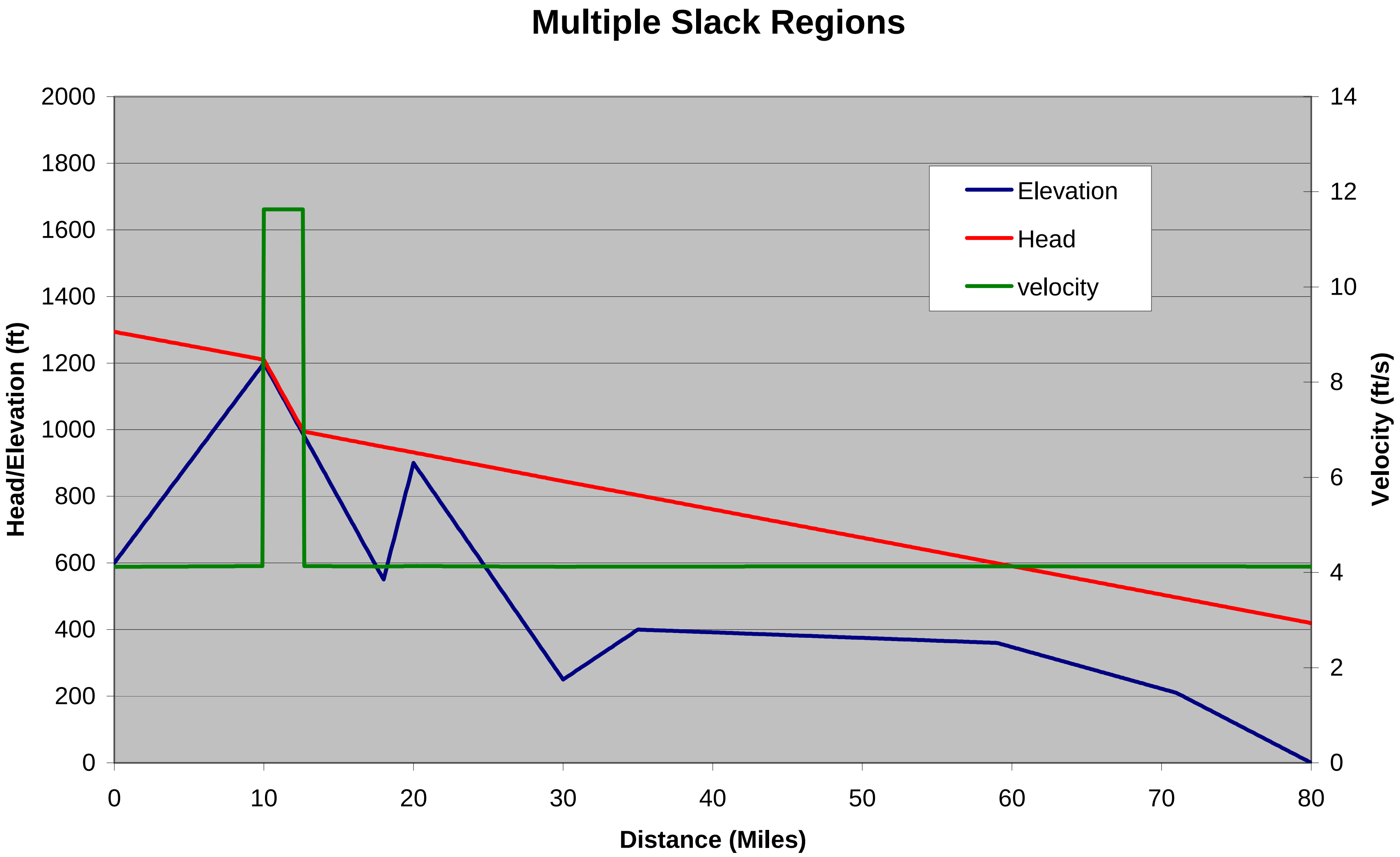

Figure 2 – Single Slack Region

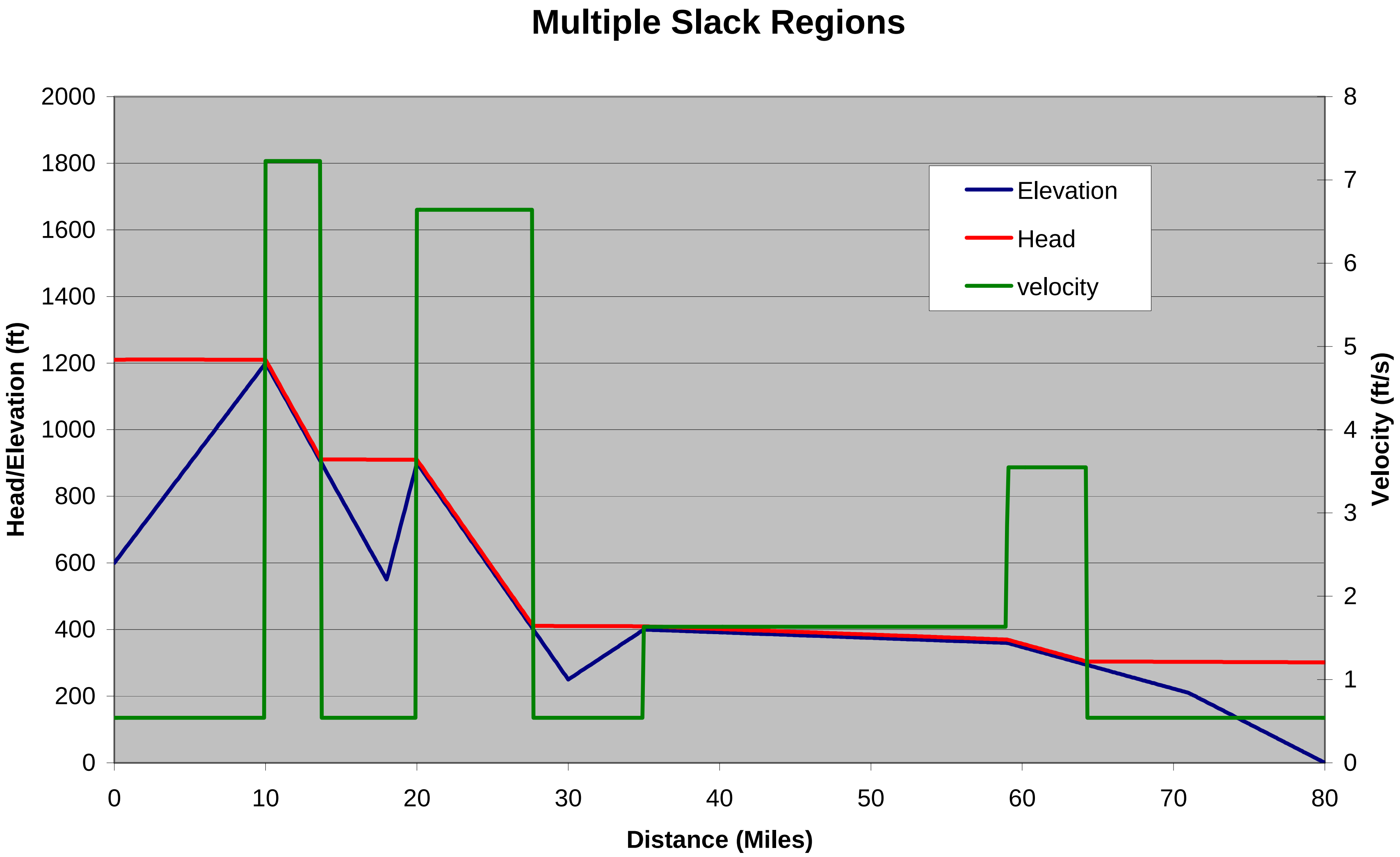

Figure 3 – Multiple Slack Regions

APPENDIX 3. SAMPLE FULL DOMAIN SOLUTION

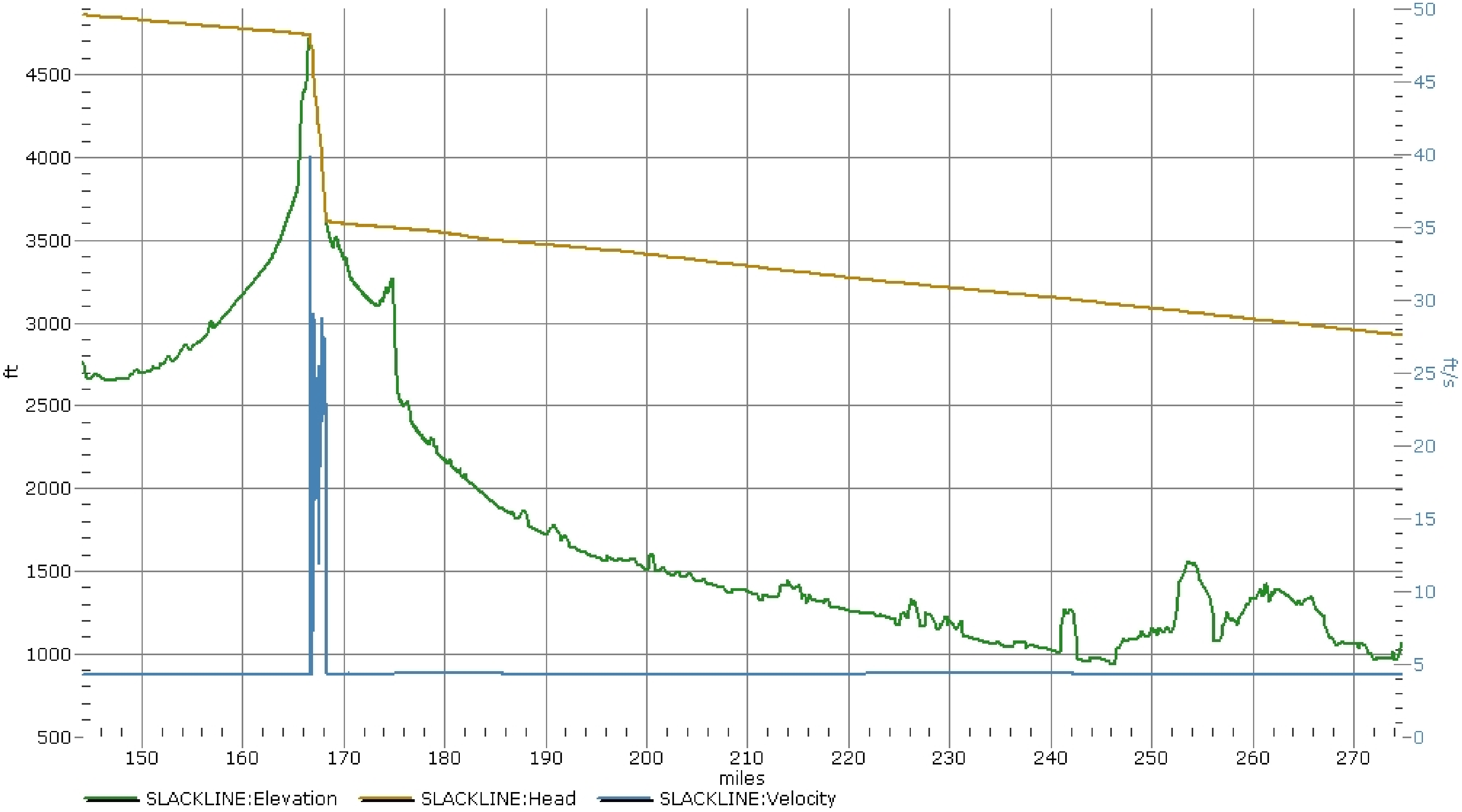

Figure 4 –Slack Region

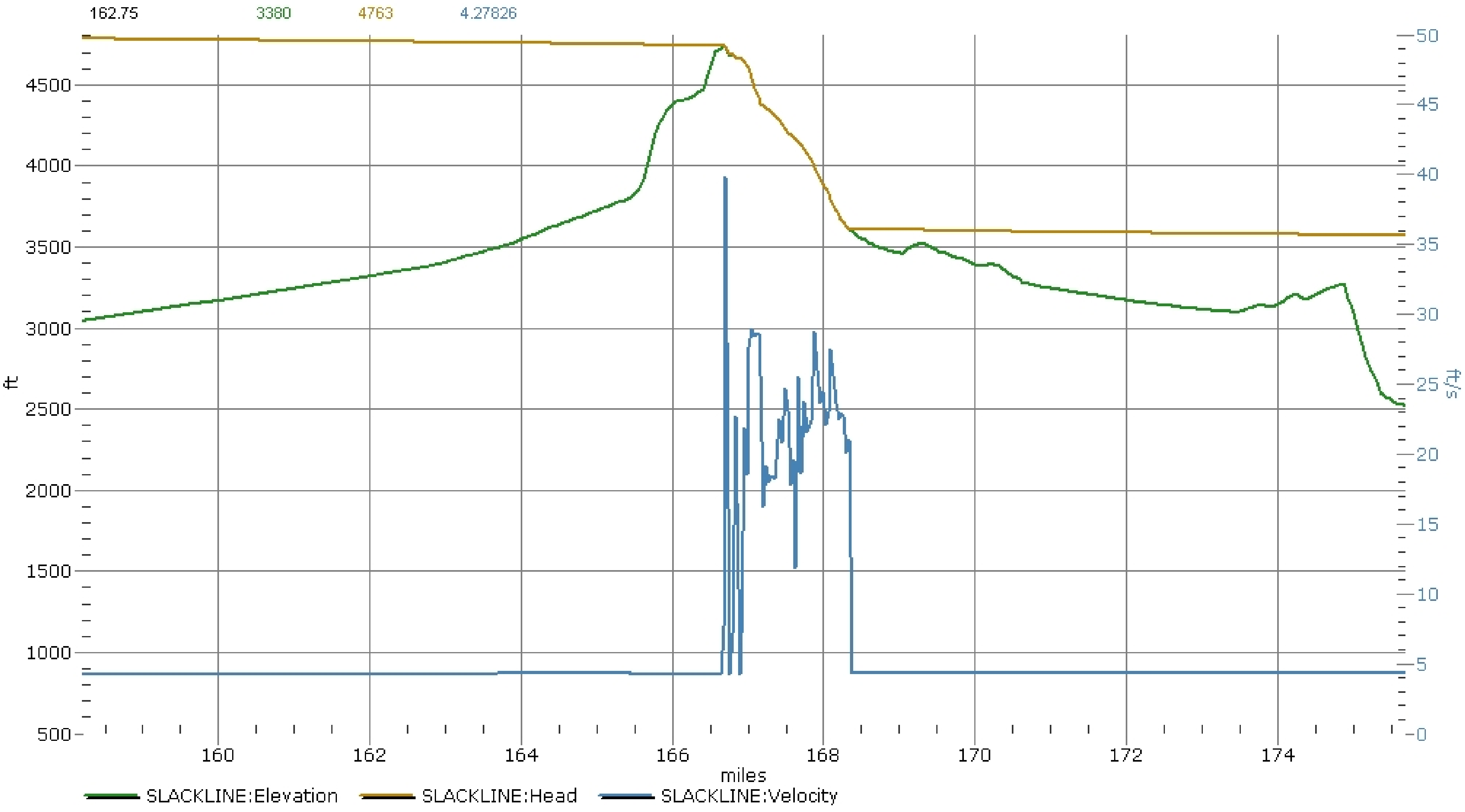

Figure 5 –Slack Region (magnified)