IMPROVING THE PERFORMANCE OF THE HARDY CROSS ALGORITHM FOR LARGE VENTILATION MODELS

*C. M. Stewart
Student, University of Queensland
Manager, Chasm Consulting
Brisbane, Australia
*(*Corresponding author: craig@chasm.com.au)*

S. M. Aminossadati,
Associate Professor, University of Queensland
Brisbane, Australia

M. S. Kizil
Associate Professor, University of Queensland
Brisbane, Australia

ABSTRACT

The Hardy Cross algorithm offers a reliable method of solving network systems of fluid flow and has become widely used for solving water and ventilation flow networks. A limitation is that computational iterations and time to solve a network rises rapidly with the size of the model and modern detailed ventilation networks have typically grown to thousands of airways. Non-linear matrix solving methods can offer improved performance, however these are more complex and may be unstable if initial estimates are poor. This paper presents improvements that can be applied to the traditional Hardy Cross algorithm to greatly reduce iterations and solving time for large ventilation models.

KEYWORDS

Gradient Method, Hardy Cross, Iterations, Network Analysis, Non-Linear, Simulation, Ventilation Modelling

INTRODUCTION

The Hardy Cross algorithm (Cross, 1936), was originally developed to solve water flow in city and urban pipe networks. Using principles of Kirchhoff's electricity current laws, Hardy Cross developed an iterative algorithm for solving non-linear equations associated with network water flow. Traditional network analysis research refers to the various members of system networks as branches, nodes and meshes. This terminology has been changed in this paper to airways, junctions and loops respectively which are terms more familiar to engineers in mine ventilation.

When Kirchhoff's laws are applied to fluid networks, the following assumptions can be made:

- The sum of fluid flows into a junction must equal the sum of fluid flows out of the junction.
- The sum of pressure losses around any loop through a network system must be equal to zero.

An iterative solution can therefore be achieved by initially guessing flow in a network, calculating the pressure losses around the network loops (Figure 1), and applying flow corrections using Newton's method for each loop if the sum of the pressure losses does not equal zero. To solve the system, a loop must be defined for every segment or series of segments between a junction of 3 or more airways.

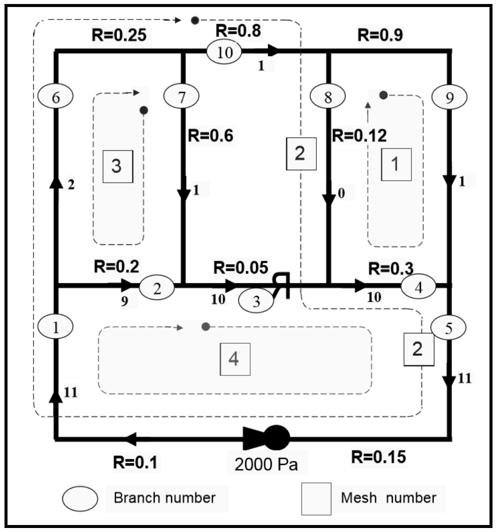

Figure 1. A typical small network consisting of loops, airways and junctions ((McPherson & Hinsley, 1993)

The Hardy Cross algorithm has been used extensively for ventilation network analysis since the introduction of the first electronic computers (Tien, 1997), however numerous other variations and methods have since been developed through the use of matrices to solve the required system of nonlinear equations. Commonly used algorithms such as the Newton Raphson method (Brown, 1969), the Gradient method (Todini & Pilati, 1988), the Linear Theory method (Gupta & Prasad, 2000), and Coupled Network Solver (G. Danko, 2008) have found general use in fluid flow network solvers (Rossman, 2000). Despite poor and sometimes unpredictable convergence performance however, the Hardy Cross algorithm still has unique properties that make it attractive for ventilation models. Some benefits are:

- Simple to program
- Resilient to poor initial guesses
- Computationally fast iterations
- Rapid convergence for mostly solved networks (Demir, Yetilmezsoy, & Manav, 2008).
- Memory efficiency
- Amenable to parallel processing due to the independent nature of each mesh loop

Causes of Poor Convergence of the Hardy Cross Algorithm

An iterative process is required for solving ventilation networks because the system of equations for airflow is based on the non-linear Atkinson's equation [$P=RQ^2$]. Non-linear equations must be linearized using the slope derivative (nRQ^{n-1}) for the algorithms to iteratively step closer to a solution. The Hardy Cross method may take hundreds or thousands of iterations to achieve an acceptable level of accuracy for larger models. Each loop is solved independently and because airways may be shared with other loops, adjusting the flow in one loop to achieve a zero-pressure balance causes imbalance in adjacent loops. Matrix methods prevent the adjacent imbalance issue by solving the loop equations simultaneously but due to the non-linear equations used may still require a dozen or more iterations to achieve acceptable accuracy. Matrix methods are more computational and memory intensive and larger systems need to use more complex "sparse matrix" mathematics and matrix re-ordering to prevent the full matrix size being re-calculated. Analysis of the potential causes of excessive iterations in Hardy Cross have suggested at least three potential reasons.

Under-Correction

The correction error can only be applied based on the linearized version of the loop equations. Because the system equations are non-linear, the corrections only step part-way towards a solution and will often under-predict the correction required, requiring more iterative steps (Figure 2).

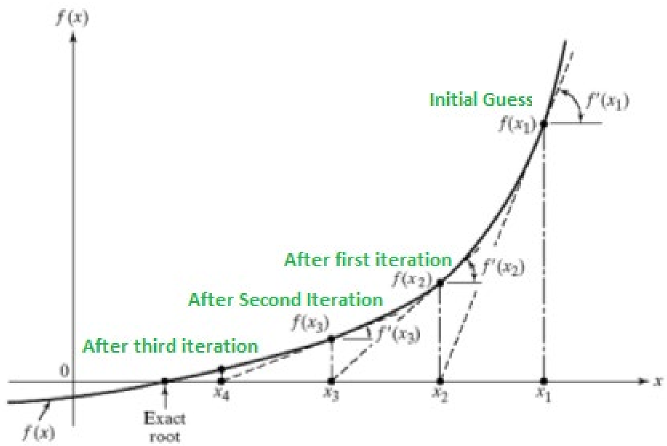

Figure 2. Convergence of Non-Linear Systems

Over-Correction

Over-correction occurs when a loop error is corrected by a large amount (often because of a poor initial guess), which greatly disrupts an adjacent loop with shared airways. The disrupted loop then also applies a large correction and propagates this into other loops resulting in an extended series of iterations to ‘repair’ the disruption. In some cases, the disruption may not be recoverable as the large corrections continue to throw adjacent loops further out of balance.

Counter-Correction

An analysis of a loop in a poorly converging network found that repeated corrections were being undone by opposite corrections from adjacent loops. During a single iteration, the cycle of corrections travelled through other adjacent loops in a network until they returned to the original loop, repeating the correction cycle. These repeating ‘harmonic’ corrections (Figure 3) were found to be the largest contributor in most models with excessive iterations. While the error correction generally gets smaller each time, the number of iterative cycles is greatly increased.

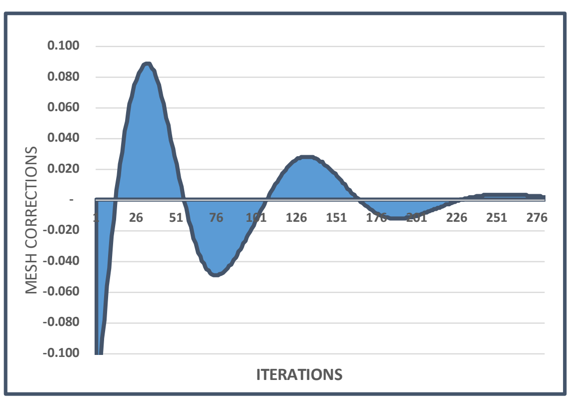

Figure 3. Loop Correction Harmonic Convergence Behavior with conventional Hardy Cross

METHODS OF IMPROVING CONVERGENCE

The following methods were proposed and developed to address the causes of excessive iterations. The methods were then rigorously tested with large numbers of randomized networks to test effectiveness.

Applying a Convergence Factor

Under-correction caused by equation linearization and the resultant ‘stepped’ error correction towards a solution can be improved by using a factor that increases the correction amount that would normally be applied to a loop by the Newton Method. Too much correction however risks divergence from the solution, increasing iterations and eventually may not be recoverable.

Limiting Correction Size

Large corrections to loops with poor initial guesses can be limited to prevent over correction effects to adjacent loops. This will minimize the effect of the correction on adjacent loops and allow the network to more gradually move towards the required solution. Restricting the over correction too much however risks additional iterations being unnecessarily required to reach a solution.

Reducing Repetitive (Harmonic) Corrections

The oscillating convergence behavior (Figure 3) of error corrections with adjacent loops resembles an oscillating mechanical system (Trigg, Rothman, Benedek, & Phillpot, 1994). To cancel the effect of harmonic oscillations in mechanical systems, either the applied force (in this case the loop corrections) or the natural frequency of the structure (in this case the network) needs to change. The order in which mesh corrections are applied dictates the direction of harmonic oscillation in a network should they occur. If every loop (mesh) correction is applied in a different order for each iteration, the flow of corrections into adjacent loops is changed and prevents the system falling into lengthy harmonic correction cycles, resulting in a solution in far fewer iterations (Figure 4). The method presented has been labelled by the author as **Iterative Random Mesh Ordering (IRMO)**.

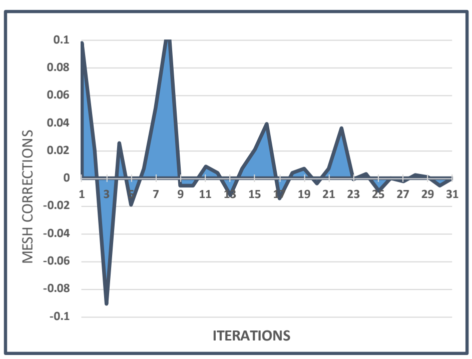

Figure 4. Loop Correction Behavior with IRMO Hardy Cross solving

TESTING OF METHODS

A method of creating randomized ventilation networks was proposed by Kim’s dissertation “An empirical study of mine ventilation network analysis methods” (Kim, 1990), where the Hardy Cross algorithm was tested against several non-linear matrix solving methods. Kim’s study concluded that despite the matrix methods requiring less iterations, the Hardy Cross solution (for ventilation networks) couple with least resistance loops remained a computationally fast method for solving ventilation networks although this was only applied to networks of up to 200 airways in size with a single fan. Sarac and

Sensogut (2000) reported similar results when comparing Hardy Cross to non-linear methods, but noted that the Hardy Cross solution slows considerably when approaching an acceptable error level.

Contemporary mine ventilation networks however have many fans and are typically up to 5000 airways or more in size (Chasm Consulting, 2015), with some models observed to exceed 50,000 branches. To test the performance of much larger networks with the Hardy Cross algorithm, Kim's software code was slightly modified to produce larger models with more fans. The effect on mesh selection was also evaluated by using both minimum sum resistance loops (which produced smaller number of airways in each loop), and a minimum resistance spanning tree method (which produced larger numbers of airways per loop).

Testing Methodology

One hundred (100) random models were generated for each size of network. The network size for each group of models was varied from 200 to 5,000 branches in size and tested against the various proposed methods, noting solving time and iterations required. Kim's randomized models were found to be typically more complex and variable and of a higher density (more junctions and less airways) than equivalent real mine examples. For example, a randomized network of 1000 airways was typically equivalent in complexity and solving time to a real mine network of several thousand airways. All models were converged to a high accuracy of 0.001 m3/s and results were averaged with the upper and lower 5% of result outliers excluded to remove excessive bias of a minority of networks from the results.

RESULTS

Kim's study demonstrated that the Hardy Cross algorithm requires increased convergence iterations with network size. Repeating the testing procedure with the increased model sizes from Kim's modified code confirmed the continuation of this trend, with the Hardy Cross routine showing exponential growth in iterations to network size. The matrix methods only showed a slight increase in iterations, however the solving time exponentially increased demonstrating no overall benefit to the Hardy Cross method using Kim's code. While Kim's matrix methods (which used dense matrix solving) could have been improved using sparse matrices, this study focused on improving only the Hardy Cross.

Iterative Randomized Mesh Ordering

Applying the IRMO method together with using a spanning tree loop selection method results in both a significant improvement in convergence iterations and demonstrates a trend that no longer increases iterations proportionally with size. Figures 5 and 6 below show iteration convergence and computational time performance to achieve mesh corrections of less than 0.001 m3/s, clearly demonstrating the superiority of using the IRMO method with spanning tree resistance loops, compared to conventional Hardy Cross with both minimum resistance and spanning loops.

The IRMO method combined with the spanning tree loops reduced iterations for large (5000 airway) randomized models from an average of 15,000 iterations to only 350, and in further testing only increased to only 500 iterations with network sizes of 15,000 branches. Networks of 500,000 airways in size were successfully tested with less than 1000 iterations. It was noted the IRMO method however showed no improvement when coupled with the minimum resistance loops. It is surmised this may be due to decreased shared airways and loop interaction in the minimum resistance loop method compared to the high number of shared airways and loop interactions in the minimum resistance spanning tree method.

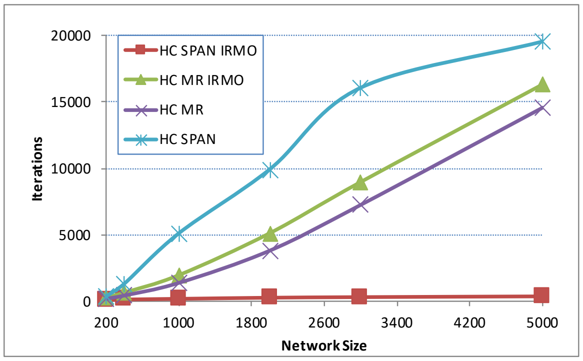

Figure 5. Convergence Iterations vs Network Size (up to 5000 branches, 0.001 m3/s error)

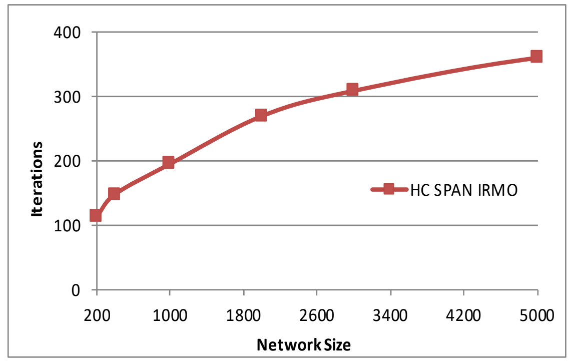

Figure 6. Convergence Iterations vs Network Size IRMO SPAN method only

Computational Speed

Computational speed is a function of the number of loops required, the number of airways included in the loops, and the number of iterations required to converge to a solution. Unlike conventional Hardy Cross where the linear growth in iterations and the growth in the number of airways combine to produce an exponential increase in solving time, the IRMO method appears to plateau convergence iterations and therefore computational speed reverts to a near linear increase proportional with the number of airways. Figures 7 and 8 demonstrate the improved behaviors of convergence in large models (up to 5000 branches, 0.001m3/s accuracy, IRMO spanning tree loop method only).

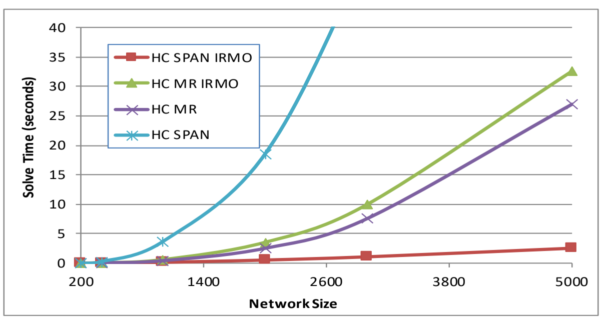

Figure 7. Computational Speed vs Network Size

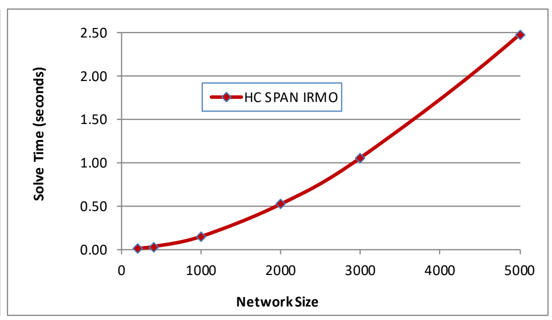

Figure 8. Computational Speed vs Network Size

Loop Correction Limiting

Correction adjustments to loops per iteration were limited to a range of values from 0.5m^3/s to 100 m^3/s. Multiple (50) randomized networks of 1000 airways were selected to form the test subjects and all tests were done with only the optimized IRMO and minimum resistance spanning tree method.

The hypothesis of this method suggests small limits risk long convergence times when a loop starts with a poor initial estimate whereas large or no limits may cause a ripple of large corrections to travel through a series of adjacent network loops and more iterations will be required to recover the solution. The results in Figure 9 showed only modest positive benefits in limiting correction size down to around 2m3/s, however the benefits faded below this and rapidly increased iteration numbers.

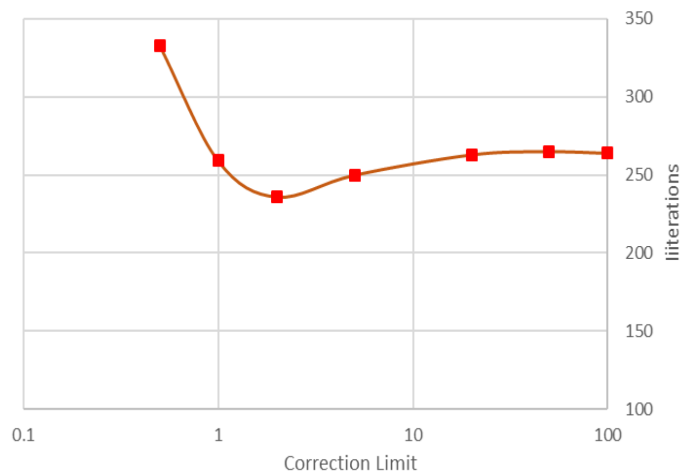

Figure 9. Correction Limiting (1000 Airways)

Correction Factors

Correction factors ranging from 0.8 (under-correction) to 1.8 (over-correction) were tested for iteration improvements. For example, the Newton method suggests that a correction of +10 m3/s needs to be applied to each airway in a loop to approximate a zero-pressure balance, then a correction factor of 1.5x would apply 15m3/s instead. The hypothesis is that correction factors of more than one (1) applied to loop corrections to each airway may allow iterations to step towards a solution faster due to skipping some subsequent intermediate correction steps. A factor that is too high may risk excessive stepping creating a diverging result with no solution.

It was found that factors up to 1.6 resulted in a 25% improvement in convergence speed. Factors or 2.0 or more occasionally did not resolve to a solution, and therefore the maximum factor used in the study was limited to 1.8. Figure 10 shows up to a 38% improvement in iteration performance compared to no adjustment to the corrections.

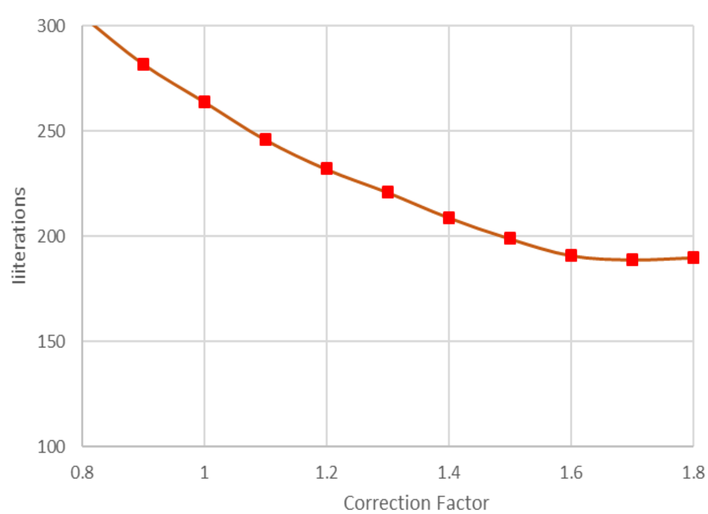

Figure 10. Correction Factor Effect (1000 Airways)

RESULT SUMMARY

Table 1 summarizes the results of the improvement strategies, with the application of the IRMO method producing the best net gain in performance improvements of the Hardy Cross algorithm.

Table 1. Results of Improvement Strategies

METHOD	POTENTIAL IMPROVEMENT	COMMENT
IRMO	100–1000%+	Higher improvement with increase model size
CORRECTION FACTOR	38%	Factor of 1.4 – 1.6 Considered Optimum
CORRECTION LIMIT	10%	Correction limit of 2m3/s gives best result

REAL WORLD PERFORMANCE

The IRMO method shows the most promise for performance increases on the randomized model testing, however limited ‘real world’ model performance testing suggests the improvement may not be as significant in many cases. The IRMO method improvements depend on the likelihood of a model falling into an error correction cycle which occurs when large numbers of shared airways are members of multiple loops resulting from a very dense models with many interconnected junctions. Real world models are generally found to be much more ordered and regular in construction than the randomized models used in this study. They also tend to be very ‘sparse’ in construction, with relatively few junctions compared to airways, and most junctions only have a maximum of three or four airway connections.

Unfortunately, there is no practical statistically correct way to compare the performance across hundreds of real-world models. Real world models are usually the intellectual properties of the companies that produce them. Comparing results to a selection of twelve (12) available real world mine ventilation models, randomly generated networks of a similar size took many more iterations to solve. The selection of real world’ models investigated ranged from 150 to nearly 10000 airways with most described as ‘very sparse’, having relatively few loops compared to the number of airways. Of the 12 real world models, 9 showed good improvements of greater than 30% reduction in iterations, while 3 of these showing extremely good improvements of 90% less iterations. Three models converged quickly anyway and showed no improvement (and no worsening) with IRMO. Applying limiting and over converging factor methods showed minor improvement in nearly all cases. Due to the limited real-world sample size, no statistical information on the proportion of improvement is presented. On average, real-world ventilation model performance is unlikely to achieve improvements shown in the random model trials however significant gains will still be made in some cases, and other types of networks (such as water or power distribution) may be more effective.

CONCLUSIONS

Iterative random mesh ordering (IRMO) coupled with mesh selection derived using a minimum resistance spanning tree method, produces greatly improved *average* convergence times and computational speed over the traditional Hardy Cross algorithm in larger randomized networks. Further improvements can be gained by increasing loop corrections by a factor of up to 1.5× or more and by limiting correction magnitudes by between 2 and 10 m^3/s. Randomized networks generated by Kim’s method however are significantly more complex than real world ventilation models and do not accurately represent typical mine networks. Limited real-world model testing shows less improvement in performance, but as demonstrated in limited real world testing, a majority of cases are likely show significant improvement.

The strengths of the Hardy Cross algorithm such as simplicity, tolerance for poor initial starting estimates and computational memory efficiency are retained, while a weakness such as convergence performance for large networks was reduced with the proposed methods. Transient simulation methods such as fire simulation which require multiple iterations of steady state simulation will also benefit in delivering faster results.

While no direct comparison was done with modern matrix solving algorithms such as the Newton Raphson Matric method, or the Modified Gradient Method, the improved methods demonstrate the traditional Hardy Cross method can potentially compete for performance in larger networks. Even if not superior in performance to modern matrix algorithms, the improved Hardy Cross method may still find opportunities where simple programming is desired, in applications or networks where matrix pre-conditioning is difficult or as a pre-conditioner for matrix methods that require better starting estimations.

REFERENCES

Brown, K. M. (1969). A quadratically convergent Newton-like method based upon Gaussian elimination. *SIAM Journal on Numerical Analysis, 6*(4), 560-569.

Chasm Consulting, C. S. (2015). Ventsim Visual User Manual V4. 320.

Cross, H. (1936). Analysis of flow in networks of conduits or conductors. *University of Illinois. Engineering Experiment Station. Bulletin; no. 286*.

Demir, S., Yetilmezsoy, K., & Manav, N. (2008). Development of a modified Hardy-Cross algorithm for time-dependent simulations of water distribution networks. *Fresenius environmental bulletin, 17*(8), 1045-1053.

G. Danko, D. B. (2008). *Application of MULTIFLUX for air, heat and moisture flow simulations*. Paper presented at the 12th U.S./North American Mine Ventilation Symposium 2008, Reno, NV.

Gupta, R., & Prasad, T. (2000). Extended use of linear graph theory for analysis of pipe networks. *Journal of hydraulic engineering, 126*(1), 56-62.

Kim, J. (1990). *An Empirical Study of Mine Ventilation Network Analysis Methods.* Pennsylvania State University.,

McPherson, M. J., & Hinsley, F. B. (1993). *Subsurface ventilation and environmental engineering* (Vol. 131): Chapman & Hall London.

Rossman, L. A. (2000). EPANET 2: users manual.

Sarac, S., & Sensogut, C. (2000). A combined method for the analysis of mine ventilation networks. *Mine Planning and Equipment Selection 2000*, 127.

Tien, J. C. (1997). Computer Simulation. *Ventilation Course Notes MST*(Chapter 12), 215-243.

Todini, E., & Pilati, S. (1988). *A gradient algorithm for the analysis of pipe networks.* Paper presented at the Computer applications in water supply: vol. 1---systems analysis and simulation.

Trigg, G. L., Rothman, S. J., Benedek, R., & Phillpot, S. R. (1994). Encyclopedia of applied physics. *Physics Today, 47*, 60.