EXPERIMENTS WITH VORONOI MESHES FOR GEOTHERMAL MODELS

Mart Baelemans[1] and Michael O'Sullivan[2]

[1] Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB, Eindhoven, The Netherlands

[2] Engineering Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

m.osullivan@auckland.ac.nz

Keywords: *Geothermal modelling, Voronoi grids, regularization.*

ABSTRACT

Various modellers have used Voronoi meshes in their geothermal models, constructed by first triangulating the region of interest and then connecting the centres of all the triangles to form a polygonal mesh. A Voronoi mesh has the nice property that the lines joining block centres are orthogonal to the block boundaries. However in some cases the Voronoi meshes used have an undesirable feature of including one or more very small sides in some blocks. In the current study we implement an algorithm for regularising Voronoi meshes, offsetting the orthogonality condition against the secondary aim of having all sides of a block being of the same size.

The method is demonstrated on a test problem of an injection/production doublet and on a mesh used in a previous modelling study.

1. INTRODUCTION

1.1 Geothermal Modelling

Geothermal modelling has become a standard tool for planning and managing the development of geothermal energy resources (e.g., O'Sullivan, et al., 2001; O'Sullivan and O'Sullivan, 2016; Popineau, et al., 2018).

The simulator most commonly used for geothermal modelling is TOUGH2 (Pruess et al., 1999), developed at Lawrence Berkeley National Laboratory. TOUGH2 is based on the finite volume method for spatial discretisation which allows the user to implement very general unstructured meshes. However some of the meshes used, with mesh refinement implemented by connecting one large block connected to two small blocks, have been poorly conditioned because the lines joining block centres are not orthogonal to the block edges. Croucher and O'Sullivan (2013) showed how the orthogonality of this type of mesh could be improved to give more accurate simulations.

Other modeller have achieved good orthogonality properties in their meshes by using the Voronoi meshes discussed below.

1.2 Voronoi meshes

A Voronoi mesh is based on a relatively simple principle, which was summarized by Aurenhammer (1991) (p.346): ”*Given some number of points in the plane, their Voronoi diagram divides the plane according to the 'nearest-neighbour rule': Each point is associated with the region of the plane closest to it*”

Voronoi meshes are constructed using a so-called Delaunay triangulation between a set of mesh nodes. This triangulation is constructed in a way that no mesh point is in the circumcircle of any of the triangles in the triangulation. The relation between the Delaunay triangulation and Voronoi diagram is shown in Figure 1.

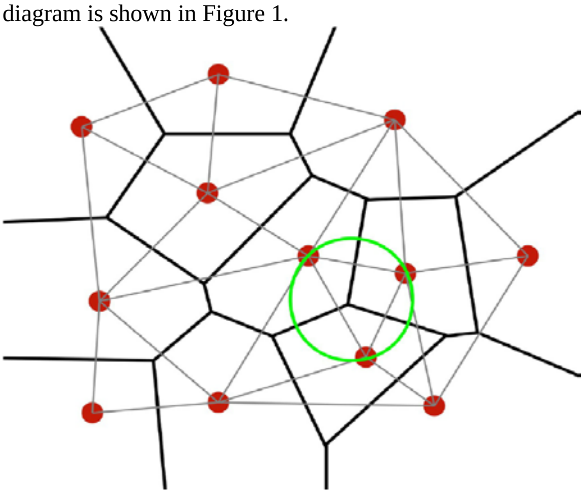


Figure 1: Example of a Delaunay triangulation (grey lines) between mesh nodes (red dots). The resulting Voronoi mesh is shown in black lines. (The circumcircle of one of the triangles is added.)

2. OPTIMIZED VORONOI MESHES

2.1 Ill-conditioned meshes

When the ratio between edge lengths is very large Voronoi elements are ill-conditioned, and can cause computational inaccuracy. This problem has appeared in many of the Voronoi meshes used for geothermal modelling in the past, e.g.: Bodvarsson et al. (1987ab); Monterrrosa (2002); Haukwa et al. (2003); Kiryukhin and Yampolsky (2004); Wu et al. (2006); Kiryukhin et al. (2007); Ketilsson et al. (2008); Montegrossi et al. (2015); Sirait et al. (2015); Kiryukhin et al. (2017).

Most of these model meshes were generated with AMESH (Haukwa, 1998) which, while guaranteeing orthogonality, may create meshes that are poorly conditioned because the sides of a block are very different in size, sometimes having one side that is very small.

An example of an ill-conditioned Voronoi element is shown in Figure 2. In this mesh, from a paper by Kiryukhin and Yampolsky (2004) the element “*bd 1*” has one edge which is significantly shorter than the others. Similarly the edge connecting element “*bx 1*” to element “*cg 1*” is very small, whereas the neighbouring elements, e.g., “*bv 1*”, do not have this problem.

Figure 3 shows another Voronoi mesh, used by Bodvarsson et al. (1987ab), which also has several ill conditioned elements (e.g., 10, 23, 25, 26).

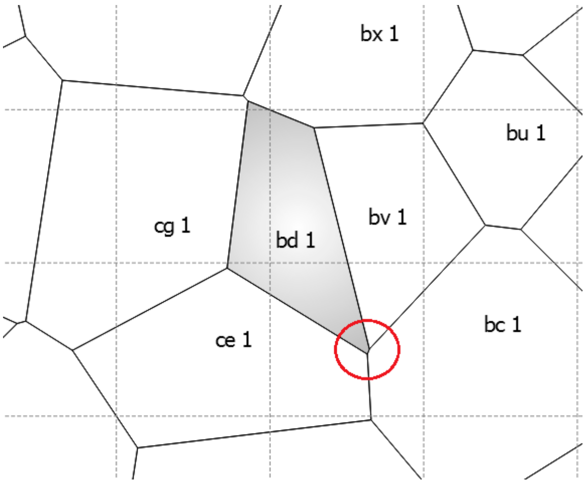


Figure 2: Example of an ill-conditioned Voronoi element with a large ratio of edge lengths (element "*bd 1*"), from Kiryukhin and Yampolsky (2004).

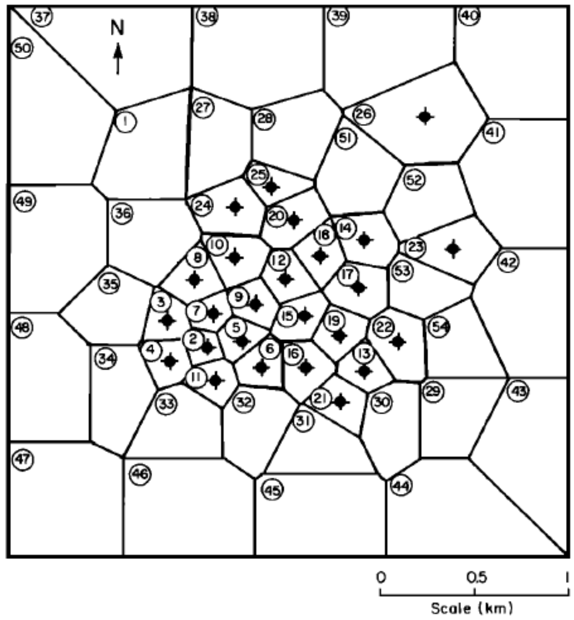


Figure 3: Another example of a Voronoi mesh with some ill-conditioned elements, from Bodvarsson et al. (1987ab).

The most common reasons for using a Voronoi grid in a geothermal model is to allow each well to be centred in its own block and to fit an irregular shaped boundary for the model. A sequence of models constructed by Kiryukhin and co-workers are of this type (Kiryukhin et al. , 2000, 2007, 2017; Kiryukhin and Yampolsky, 2004; Kiryukhin and Vereina, 2005.

Models with Voronoi grids were used in studies of a nuclear waste repository site, proposed for Yucca Mountain, in order to follow a complex geological structure (e.g., Haukwa et al., 2003; Wu et al., 2006)

However as with other models, with irregular unstructured grids, it is difficult to deal with anisotropic permeability in models based on a Voronoi grid. The test cases discussed below all use isotropic permeability.

2.2 Regularisation techniques

A number of methods exist for improving the quality of Voronoi meshes. In the study discussed here three different mesh regularisation methods were investigated. The first and main optimization method applied was that developed by Sieger et al., (2010). Also an attempt was made to apply two optimization methods available in the PyTOUGH tool kit (Croucher, 2011, 2015) for regularising Voronoi meshes containing "bad elements".

In the paper by Sieger et al., (2010), a 2D mesh improvement technique is presented for optimising Voronoi meshes used in finite element computations. The method optimizes the mesh by minimizing a mesh "energy" that depends on the shape of the triangles in the underlying Delaunay triangulation. Minimizing the energy will generally reduce the short edges in a Voronoi mesh.

The mesh energy as defined by Sieger et al. (2010) is calculated as follows:

$$E(\mathbf{v}_1,...,\mathbf{v}_m)=\frac{1}{2}\sum_{all\ \tau} R_\tau (R_\tau - 2r_\tau) \qquad (1)$$

In (1), E is the mesh energy, $\mathbf{v}$ is a mesh node (Delaunay vertex), R_τ and r_τ are the circumradius and inradius of a triangle. This definition of the energy actually does nothing more than sum the squares of distances between incenter and circumcenter for each triangle. For an ideal equilateral triangle, this distance is zero (Figure 4).

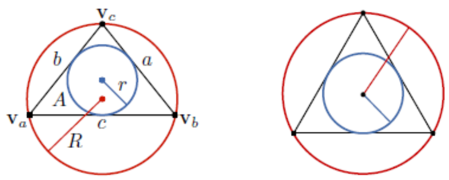


Figure 4: Left: A triangle ($\mathbf{v}_a$; $\mathbf{v}_b$; $\mathbf{v}_c$) with circumradius *R* and inradius *r*. Right: For an equilateral triangle circumcenter and incenter coincide (taken from Sieger et al., (2010)).

Using this simple expression for the energy, a gradient of the energy can be calculated for each node in the mesh analytically. The energy can be minimized by iteratively moving all nodes in the direction of their negative energy gradient. A step is only made if this actually reduces the mesh energy. The step size is reduced and the energies are again compared. Because all nodes are moved at the same time, this can locally increase the energy and thus reduce the quality of the mesh.

In the paper by Sieger et al. (2010) the analytical expression for the energy gradient and a proposed algorithm to implement this optimization is provided. The paper also contains a clear explanation of how to apply the optimization technique.

The optimization routine can iteratively be repeated. After each iteration, the Delaunay property is re-established before the energy and its gradient are recalculated.

For the implementation of the Sieger energy optimization method on geothermal Voronoi meshes a python routine was developed. As discussed in the Appendix a few minor changes had to be made to the algorithm suggested by Sieger et al. (2010) to implement it satisfactorily.

The python code and a user manual are available from the first author (m.g.j.baelemans@student.tue.nl).

3. MESH REGULARISATION RESULTS

The application of the mesh energy optimization (called here "MeshEnOpt") technique (Sieger et al., 2010) to two model meshes is discussed here. The first model is a 2D single layer model of a production/injection doublet system. The mesh shown in Figure 5 was set up with the commercial code PetraSim (PetraSim, 2018). The corresponding meshes after 1 and 10 iterations of MeshEnOpt are shown in Figures 6 and 7, respectively. By eye the quality of the mesh has clearly been improved. The plot of the total mesh energy vs iteration number in Figure 8 confirms the improvement and shows that after 10 iterations there is little further improvement in the mesh.

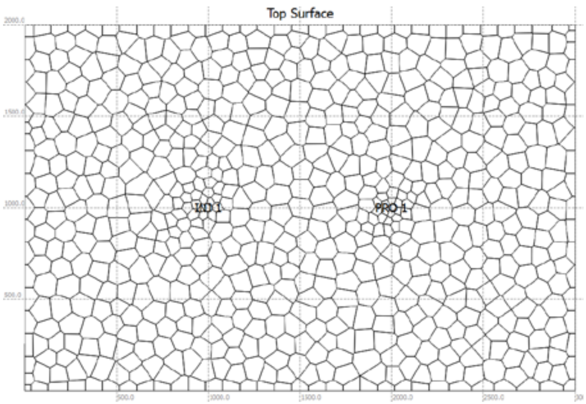


Figure 5: Petrasim mesh for the doublet problem

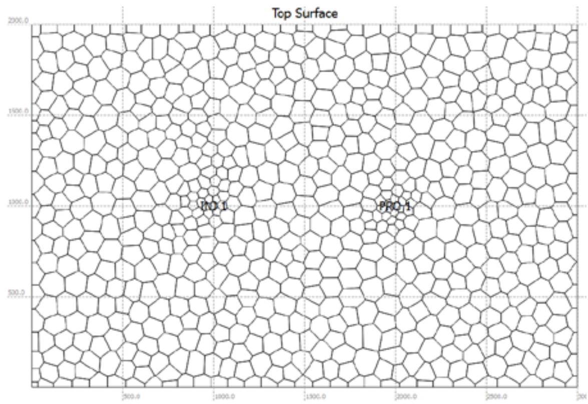


Figure 6: Petrasim mesh for the doublet problem after one iteration of mesh energy optimization

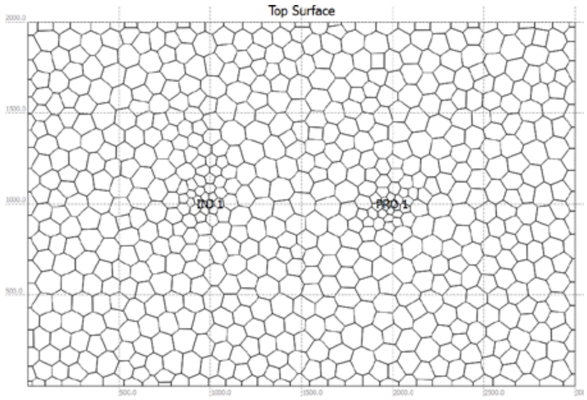


Figure 7: Petrasim mesh for the doublet problem after ten iterations of mesh energy optimisation

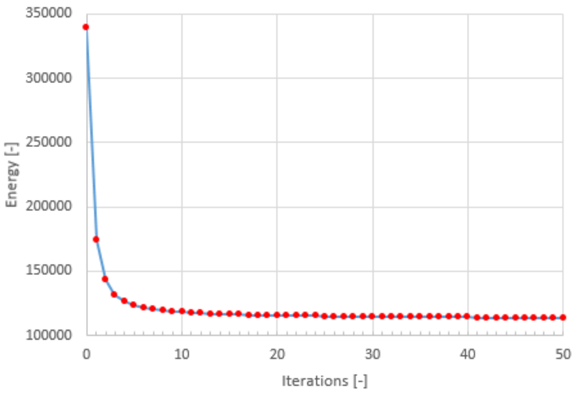


Figure 8: Doublet mesh energy vs iteration number

The second model considered is one of Pauzhetsky geothermal field, Kamchatka, Russia discussed by Kiryukhin and Yampolsky (2004). The most important differences between this mesh, shown in Figure 9, and the doublet mesh are the number of column centres that are fixed in position because they correspond to the locations of wells (32 of the 98 column centres) and the small number of columns (only 98 compared to 706 in the doublet mesh).

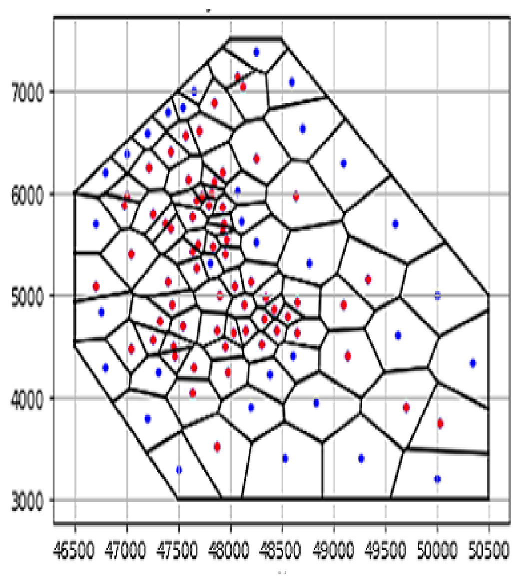


Figure 9: Original mesh for the Pauzhetsky model, from Kiryukhin and Yampolsky (2004). Fixed column centres are shown by red dots.

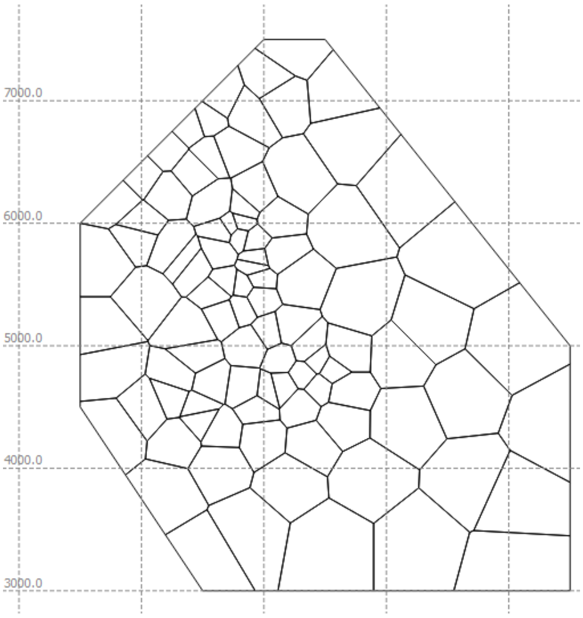


Figure 10: Mesh for the Pauzhetsky model after 5 iteration of mesh energy optimisation

By eye, there is some improvement in mesh quality but it is less impressive than for the doublet mesh. As shown in Figure 11, the mesh energy is not reduced further after 5 iterations.

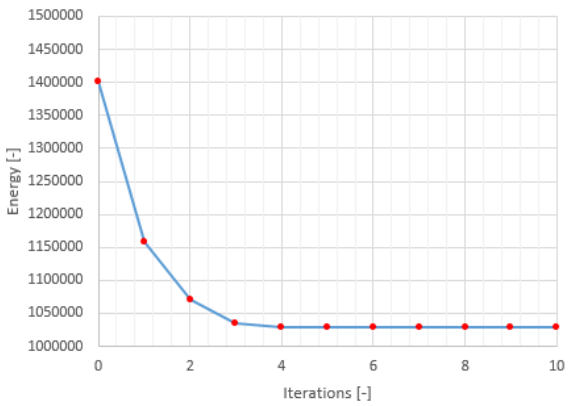


Figure 11: Mesh energy vs iteration number for the Pauzhetsky model

A comparison of the performance of MeshEnOpt for the two models is given in Table 1. The improvement factor (ratio of initial over final energy) is clearly much better for the doublet mesh, confirming the by-eye improvement.

Table 1: Comparison of mesh optimisations

Quantity	Doublet model	Pauzhetsky model
Number of columns	706	98
Fixed columns	2	32
Initial mesh energy	3.4E5	1.4E5
Final mesh energy	1.15E5	1.03E5
Improvement factor	2.96	1.36
Number of iterations	10	5

4. SIMULATION RESULTS

The next stage in the investigation was test the effect of mesh optimisation on simulation results.

Table 2: Model parameters

Parameter	Value
Rock density	2500 kg/m^3
Porosity	0.1
x,y permeability	2.0E-13 m^2
Thermal conductivity	2.5 W/m K
Specific heat of rock	1000 J/kg K
Initial pressure	4 MPa
Initial vapour saturation	0.5
Initial temperature	250.33 °C
Injection rate	5 kg/s
Injection enthalpy (20 °C)	83.9 kJ/kg
PI for production well	1.0E-10
Deliverability pressure	3.6 MPa
Relative permeability curves	Linear X-curves
Immobile water, fully mobile steam	S_v=0.5
Fully mobile water, immobile steam	S_v=0.0

A two-phase version of the doublet problem was selected as the test case as the movement of phase change and thermal fronts through the model is a severe test of the numerical accuracy. Some model parameters are given in Table 2.

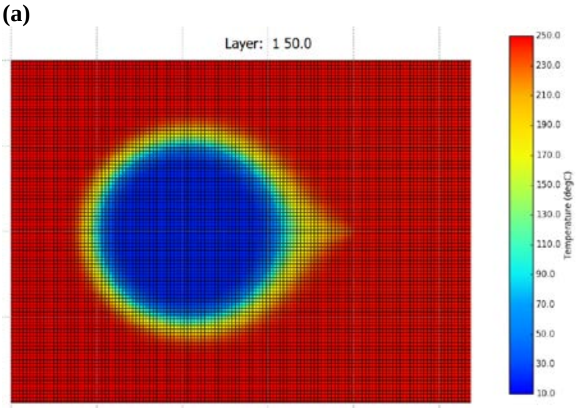
As a base case the problem was solved on a very fine rectangular mesh (20 m x 20 m blocks, covering a region 3 km x 2 km) and using high accuracy 9-point differencing (Forsythe and Wasow, 1960; Pruess and Bodvarsson, 1983). The parameters used in the model are given in Table 2. In all cases AUTOUGH2 (Yeh et al., 2012), the University of Auckland's version of TOUGH2 (Pruess et al., 1999), was used as the simulator.

The same problem was then run again using the original PetraSim mesh (Figure 5) and the optimised mesh after 10 iterations of MeshEnOpt (Figure 7). Results for the temperature distribution at 40 years are shown in Figure 12 (a), (b) and (c) for the fine mesh, original mesh and optimised mesh, respectively.

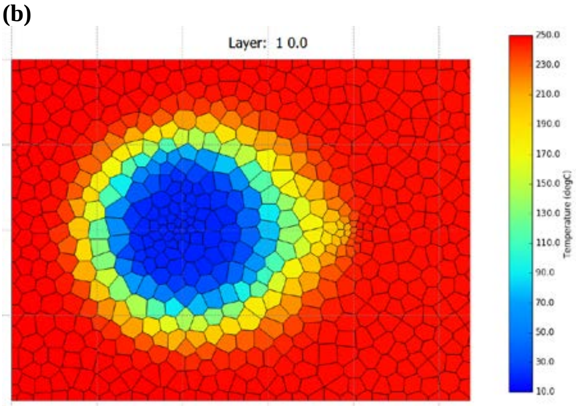
By eye the results from the optimised mesh are more comparable with the fine rectangular mesh (which was assumed to be the most accurate), but both Voronoi meshes produce results that smear out the thermal front considerably. This is to be expected with coarser meshes as the upstream weighting in TOUGH2 is known to cause numerical dispersion (e.g., O'Sullivan and Pruess, 1980; Croucher et al., 2004; Oldenburg and Pruess, 2000).

Results for the vapour saturation at 5.35 years are shown in Figure 13. Again the phase-change front is smeared out more in the coarser Voronoi meshes.

(a)



(b)



(c)

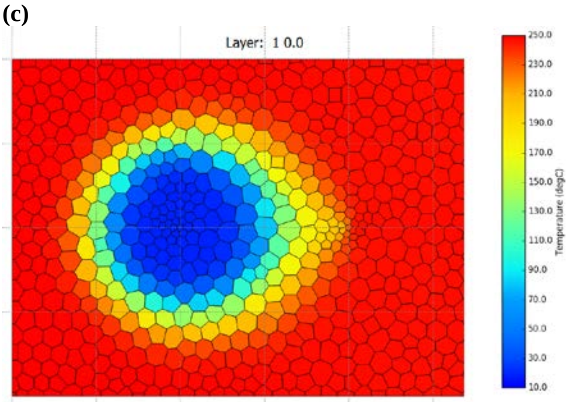
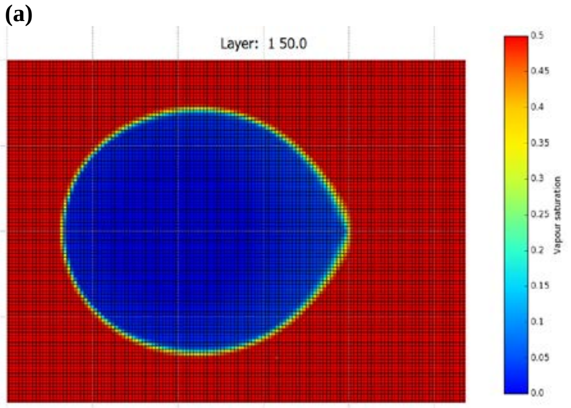
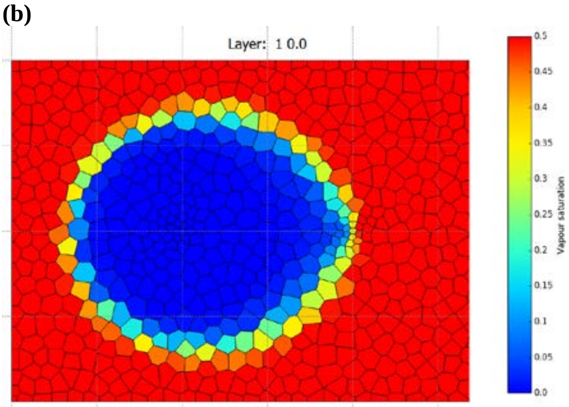


Figure 12: Temperature distribution at 40 years, (a) Fine rectangular mesh, (b) Original Voronoi mesh, (c) Optimised Voronoi mesh.

(a)



(b)



(c)

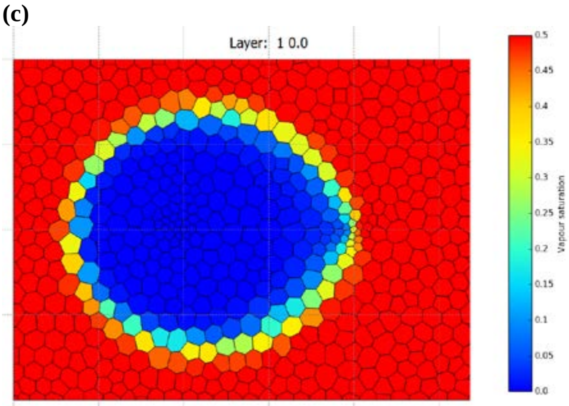


Figure 13: Vapour saturation distribution at 5.35 years, (a) Fine rectangular mesh, (b) Original Voronoi mesh, (c) Optimised Voronoi mesh.

To compare the results more quantitatively plots of the temperature and vapour saturation at x=1500 m and y= 1500 m were made (see Figures 14 and 15, respectively). Note that the injection well is located at x=1000 m and y= 1000 m and the production well is at x=2000 m and y= 1000 m.

The results from the optimised Voronoi mesh are not clearly better than the results for the original Voronoi mesh. For vapour saturation (Figure 15) the results for the optimised mesh are better but for temperature (Figure 14) they are worse.

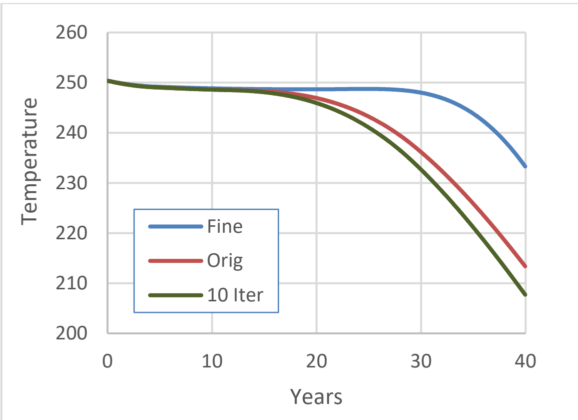


Figure 14: Temperature vs time at (1500, 1500). Fine rectangular mesh, Original Voronoi mesh, Optimised Voronoi mesh after 10 iterations

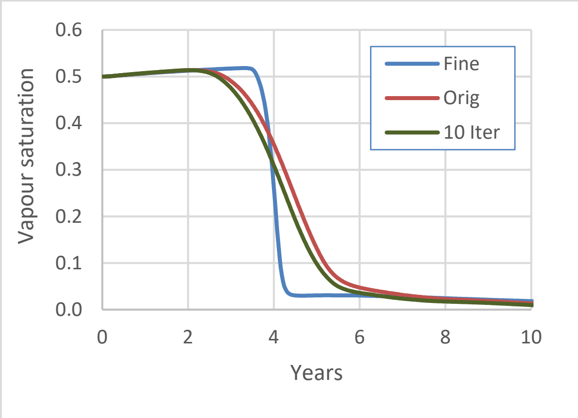


Figure 15: Vapour saturation vs time at (1500, 1500). Fine rectangular mesh, Original Voronoi mesh, Optimised Voronoi mesh after 10 iterations

A plot of steam flow from the production well, given in Figure 16, shows that the results from the original and optimised Voronoi meshes are indistinguishable, but both differ from the fine mesh results.

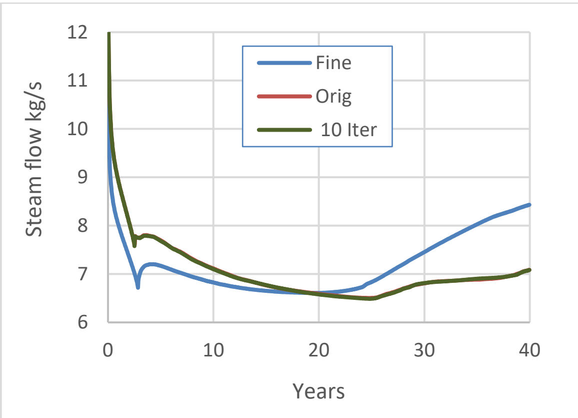


Figure 16: Steam flow from the production well vs time. Fine rectangular mesh, Original Voronoi mesh, Optimised Voronoi mesh after 10 iterations

CONCLUSIONS

The MeshEnOpt method we have implemented is very effective in optimising Voronoi meshes. The optimised meshes look nicer and potentially should produce more accurate simulations, however for the doublet test problem considered here the results from the optimised mesh were not significantly different to those from the original mesh.

Further testing on models of real geothermal fields is required to determine whether or not it is worth making the effort to optimise Voroni meshes.

REFERENCES

Aurenhammer. F.: Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), pp. 346–405. (1991).

Bodvarsson, G.S., Pruess, K., Stefansson, V., Bjornsson, S., Ojiambo, S.B.: East Olkaria Geothermal Field, Kenya 1. History Match with Production and Pressure Decline Data. *Journal of Geophysical Research*, 92(B1), pp. 521-539. (1987a)

Bodvarsson, G.S., Pruess, K., Stefansson, V., Bjornsson, S., Ojiambo, S.B.: East Olkaria Geothermal Field, Kenya 2. Prediction of Well Performance reservoir Depletion. *Journal of Geophysical Research*, 92(B1), pp. 541-554. (1987b).

Croucher, A.E., O'Sullivan, M.J., Kikuchi, T., Yasuda, Y.: Eulerian-Lagrangian tracer simulation with TOUGH2. *Geothermics*, 33(4), pp. 503-520. (2004).

Croucher, A.E.: PyTOUGH: a Python scripting library for automating TOUGH2 simulations, *Proc. 33rd New Zealand Geothermal Workshop*, Auckland, New Zealand. (2011).

Croucher, A. E., O'Sullivan, M. J.: Approaches to local grid refinement in TOUGH2 models. *Proc. 35th New Zealand Geothermal Workshop*. Rotorua, New Zealand. (2013).

Croucher, A. E.: Recent developments in the PyTOUGH scripting library for TOUGH2 simulations. *Proc. 37th New Zealand Geothermal Workshop*. Taupo, New Zealand. (2015).

Forsythe, G. E., Wasow, W. R.: *Finite-Difference Methods for Partial Differential Equations*. John Wiley. New York. (1960).

Haukwa, C.: *AMESH, a mesh creating program for the integral finite difference method: A User's Manual*, Report LBNL-45284, Earth Sciences Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, USA. 53 p. (1998).

Haukwa, C.B., Wu, Y-S., Bodvarsson, G.S.: Modeling thermal–hydrological response of the unsaturated zone at Yucca Mountain, Nevada, to thermal load at a potential repository. *Journal of Contaminant Hydrology*, 62–63, pp. 529–552. (2003).

Ketilsson J., Axelsson, G., Palsson, H., Jonsson, M.T.: Production capacity assessment: numerical modeling of geothermal resources. *Proceedings, 33rd Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California. (2008)

Kiryukhin, A.V., Pruess, K., Maltseva, K., Delemen, I., Tisheko, Y.: Modeling studies of the Paratunsky geothermal field, Kamchatka, Russia. *Proceedings, 25th Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California. (2000).

Kiryukhin, A.V., Yampolsky, V.A.: Modeling study of the Pauzhetsky geothermal field, Kamchatka, Russia. *Geothermics*, 33(4), pp. 421–442. (2004).

Kiryukhin, A.V., Vereina, O.B..: Modeling of the fault type geothermal reservoir (Dachny site, Mutnovsky geothermal field. *Proceedings, 30th Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California. (2005).

Kiryukhin, A.V., Asaulova, N.P., Rychkova, T.V., Obora, N.V., Manukhin,Y.F., Vorozheikina, L.A.: Modeling and forecast of the exploitation the Pauzhetsky geothermal field, Kamchatka, Russia. *Proceedings, 32nd Workshop on Geothermal Reservoir Engineering,* Stanford University, Stanford, California. (2007).

Kiryukhin, A.V., Vorozheikina, L.A., Voronin, P.O., Kiryukin, P.A.: Thermal and permeability structure and recharge conditions of the low temperature Paratunsky geothermal reservoirs in Kamchatka, Russia. *Geothermics*, 70, pp. 47-61. (2017).

Montegrossi G., Pasqua C., Battistelli A., Mwawongo G., Ofwona C.: 3D Natural State Model of the Menengai Geothermal System, Kenya. *Proc. World Geothermal Congress 2015*, Melbourne, Australia. (2015).

Monterrosa, M.E.: Reservoir modelling for the Berlin geothermal field El Salvador. *Proceedings, 27th Workshop on Geothermal Reservoir Engineering,* Stanford University, Stanford, California. (2002).

Oldenburg, C.M., Pruess, K.: Simulation of propagating fronts in geothermal reservoirs with the implicit Leonard total variation diminishing scheme. *Geothermics*, 29, pp. 1-25. (2000).

O'Sullivan, M.J., Pruess, K.: Analysis of injection testing of geothermal reservoirs. *Transactions Geothermal Resources Council*, 4, pp. 401-404. (1980).

O'Sullivan, M.J., Pruess, K., Lippmann, M.: State-of-the-art of geothermal reservoir simulation. *Geothermics*, 30 (4), 395–429. (2001).

O'Sullivan, M. J., and O'Sullivan, J.P.: Reservoir Modeling and Simulation for Geothermal Resource Characterization and Evaluation. In: R. DiPippo (Ed.), *Geothermal power generation: developments and innovation*. Woodhead Publishing. (2016).

PetraSim: *PetraSim 2018 User's Manual*. Thunderhead Engineering, Kansas, USA. Pp. 1-137. (2018)

Popineau, J., O'Sullivan, J. P., O'Sullivan, M. J., Archer, R., and Williams, B.: An integrated Leapfrog/TOUGH2 workflow for geothermal production modelling. In *7th African Rift Geothermal Conference*. Kigali, Rwanda. (2018).

Pruess, K., Bodvarsson, G. S.: A seven-point finite difference method for improved grid orientation performance in pattern steam floods. *Paper SPE-12252, presented at the Seventh Society of Petroleum Engineers Symposium on Reservoir Simulation, San Francisco, CA, U.S.A.* (1983).

Pruess, K., Oldenburg, C., Moridis, G.: *Tough2 User's Guide, Version 2.0*. Lawrence Berkeley National Laboratory, Berkeley, California. (1999).

Sieger, D., Alliez, P., Botsch, M.: Optimizing Voronoi diagrams for polygonal finite element computations. *Proceedings of the 19th International Meshing Roundtable, IMR 2010*, pp. 335–350. (2010).

Sirait, P., Ruly H. Ridwan, R.H., Battistelli, A.: Reservoir Modeling for Development Capacity of Dieng Geothermal Field, Indonesia. *Proceedings, 40th Workshop on Geothermal Reservoir Engineering,* Stanford University, Stanford, California. (2015).

Wu, Y-S., Mukhopadhyay, S., Zhang, K., Bodvarsson, G.S.: A mountain-scale thermal–hydrologic model for simulating fluid flow and heat transfer in unsaturated fractured rock. *Journal of Contaminant Hydrology,* 86(1-2), pp. 128–159. (2006).

Yeh, A., Croucher, A.E., O'Sullivan, M.J.: Recent developments in the AUTOUGH2 simulator. *Proc. TOUGH Symposium 2012*, Berkeley, California. (2012).

APPENDIX

For the implementation of the Sieger mesh energy optimization technique on geothermal Voronoi meshes made by AMESH two issues required extra attention.

(i) The Voronoi meshes discussed in Sieger et al. (2010) always have some nodes which become column centres in the reservoir mesh on the domain boundary. This is generally not the case for Voronoi meshes created by AMESH. In the Python routine created to optimize geothermal reservoir meshes a check is built in to decide if boundary nodes are on the actual boundary of the domain or not. If they are not, the contribution of the boundary triangles to the energy sum is no longer doubled (as is advised by Sieger et al. (2010)) but the movement of those nodes is limited to be towards the boundary. The reason for this constraint is that in general the optimization routine tends to pull boundary nodes inwards, reducing triangle sizes and thus distances between incenter and circumcenter.

(ii) Secondly the triangulation used by Sieger et al. (2010) is generated using a C++ software tool called "The Computational Geometry Algorithms Library". In the Python code developed in this work, AMESH is used for the generation of Voronoi meshes. Unfortunately the underlying Delaunay triangulation is not included in the output of AMESH. Thus to use the optimization technique in combination with AMESH the triangulation had to be reverse engineered from the Voronoi mesh. This is by far the most computational expensive step of the routine. In every iteration of the optimization routine AMESH is used to create a new optimized mesh (and re-establish the Delaunay property of the triangulation).