When an exact formula is not enough: the counter-intuitive nature of gas flow in anisotropic porous media

by Michelle Boham

B.Sc. Chemistry, Thompson Rivers University, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE IN ENVIRONMENTAL SCIENCE

In the Department of Mathematics and Statistics

Thesis Examining Committee:

Yana Nec (PhD), Associate Professor and Thesis Supervisor, Department of Mathematics and Statistics, Thompson Rivers University

Heidi Huttunen-Hennelly (PhD), Associate Professor and Committee Member, Department of Physical Sciences (Chemistry), Thompson Rivers University

Mark Paetkau (PhD), Teaching Professor and Committee Member, Department of Physical Sciences (Physics), Thompson Rivers University

Rebecca Tyson (PhD), Full Professor and External Examiner, Department of Mathematics, The University of British Columbia - Okanagan Campus

April 2023 Thompson Rivers University

Abstract

Landfills are the most common method of waste management. Inside a landfill facility anaerobic waste decomposition releases gases – including methane and carbon dioxide – which can escape into the environment when mismanaged. Methane in sufficiently high concentrations is explosive, flammable, toxic, and is one of the major contributors to greenhouse gases. Therefore, it is vital to know how the gas flows within the solid waste matrix to allow for adequate collection, safe removal and control of emissions. In particular, a correctly working facility would aim to collect as much gas as possible to prevent inadvertent escape into the environment. Current practices rely on decades of experience; however, there is little to no mathematical analysis readily available. Designers and operators require access to gas flow information to build and manage landfills with efficiency and minimal environmental impacts.

One of the important effects is the intrinsic heterogeneity of the waste matrix. Such types of media are known as anisotropic, since the heterogeneity means the resistance to fluid flow depends on the flow direction. Analytical solutions exist for anisotropic media, describing the fluid flow by giving pressure and velocity at any desired point in simplified geometric settings. However, since the pressure gradient (velocity) varies over four orders of magnitude from the perimeter of the landfill to the centre, traditional methods of visualization, such as isocontours for pressure or arrow representation for the velocity vector field, are ineffective. A custom code was developed in Octave/MATLAB to visualize a large set of configurations with an arbitrary number of sectors and distinct permeabilities. This code implements backward integration of pathlines with customized seeding of starting points and is focused on the ability to accurately locate any present stagnation points and visualize how the gas moves around them.

The code's robustness was verified by running it for a wide range of sector angles and landfill parameters. The success of these runs was then classified by the qualitative visual features present in each flow field, such as reasonable density, smoothness of pathlines, and efficient coverage of key points in the flow. The code was only deemed robust and ready to use by practitioners once it was able to integrate and plot pathlines for a wide range of parameters. These visuals will provide landfill designers and engineers with crucial information regarding the efficiency of landfill operations. Specifically, the code will make it possible to identify regions of recirculation and escape, subsequently aiding in the design and optimization of more efficient and environmentally conscious landfills.

Keywords: analytical solutions, anisotropic media, isocontours, landfill gas, landfill well, pathlines, permeability, vectors

Contents

Acknowledgements

First and foremost, I cannot begin to express my appreciation and gratitude to my supervisor and mentor Yana Nec. Your endless support and guidance made this project possible, and I am truly honoured to have been given the privilege of learning from you and being one of your students.

I would also like to thank Yana for the graduate scholarship you gave me from your research grants and for nominating me for the Telus Excellence in Science Award. I would also like to thank the donors of the Telus Award for choosing me as the recipient for your generous award.

Finally, I would like to express my deepest appreciation to my current and previous committee members, Mark Paetkau, Heidi Huttunen-Hennelly and Richard Taylor for your guidance and feedback.

List of Figures

List of Tables

1 Introduction

Historically landfills have been the most common method of waste management. During the operating life of a landfill facility waste is periodically added and over time it decomposes and settles. This process forms layers as well as laterally distributed subdomains, each with a distinct density and permeability which depends on the composition of each layer. As a direct result of the decomposition reaction, a mix of primarily methane and carbon dioxide is generated via the anaerobic decomposition of the solid waste material. Both methane and carbon dioxide pose various health and safety concerns. Methane and carbon dioxide are major contributors to greenhouse gases, however methane has a higher heat trapping ability and is also flammable (Wang et al., 2022). Therefore when left to accumulate, methane has the potential to lead to fires or even explosions (Li et al., 2012).

To remedy this issue, collection wells are installed within the landfill with the purpose of drawing out the gas, preventing it from being released into the environment. These wells are constructed in various formations, e.g. horizontal, vertical and combined networks, and they utilize a vacuum system to direct the gas to a collection point (Li et al., 2012). Figure 1 shows a schematic of a landfill gas collection well: landfill well embedded within solid waste (left) and a two-dimensional cross-section (right). Although this diagram shows a vertical well, the same structure applies to a horizontal well. The far end of the well is blocked, and suction is applied at the outlet. The gas can enter the well at any point along the pipe via small perforations in the pipe wall. Therefore the pressure in the well is lowest near the outlet (left panel, top cross-section) and the vacuum slowly dissipates towards the far end of the well (left panel, bottom cross-section). Thus taking a single two-dimensional cross-section with a suction level in the full range between the two extreme values is the setting being studied in this project. In other words, a two dimensional section with a given suction value can represent a position anywhere along the well with different outlet vacuum. In light of the above, all gas flow demonstration below will be done on a two-dimensional cross-section.

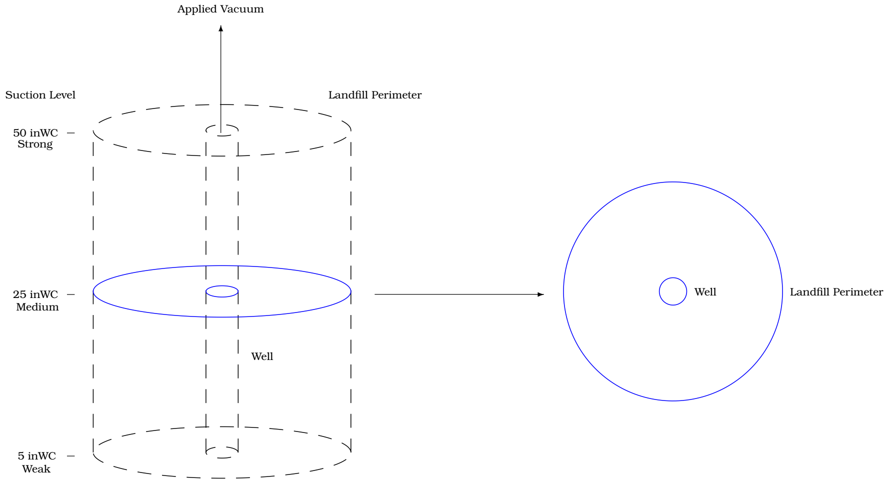


Figure 1: Simplified three dimensional and cross section examples of a generic landfill set-up. Note: In a physical landfill wells can be horizontal, vertical or a combination network. The dimensions in this figure are not to scale and realistic suction variation is non-linear.

If the collection system fails, the escaping gases can lead to the catastrophic events mentioned above. Therefore when designing a landfill, it is important to know how the gas is moving, what causes it to escape and where that might happen. Unfortunately, despite the fact that landfills play a vital role in today's society, there is surprisingly little research, and education focused on the mechanisms, operation, and management of landfills. By contrast, decomposition chemistry, toxic by-products and the environmental impact thereof are well researched. With current technology the movement of landfill gas is very difficult to monitor, so analytical solutions were previously obtained to map the gas movement surrounding the landfill wells (Nec and Huculak, 2020). Although the solutions are analytical and in closed mathematical form, they are expressed via complicated formulae and are impossible to interpret without additional visualization aids. Therefore the purpose of this project was to develop an accessible and versatile code, which allows users to visualize the gas flow for various landfill settings. Availability of such visualization aids will facilitate the advancement of engineering practices and decision making in regards to the well's ability to collect gas.

1.1 Landfill Operation and Parameters

A landfill is essentially a waste management tool, like any other tool or machine, where different operational settings and variables affect the efficiency of operations. The primary variables responsible for the operation of a landfill are well suction, landfill radius, gas generation rate and the permeability of the waste matrix. Well suction is the amount of vacuum applied to the well and is recorded in inches of water column (inWC), defined as the pressure exerted by a column of water that is one inch high. This setting is the easiest to change and is controlled by the landfill operators. Figure 1 shows how the suction level varies along the well when moving away from the vacuum source. The landfill radius is the distance from the centre of the well to the perimeter of the landfill. The generation rate is the amount of gas produced per unit time as the waste decomposes. This variable is highly dependent on the type of waste. For example, waste that decomposes easily, such as agricultural organic material or food waste, will generate more reaction by-products than construction and demolition waste. Finally, the permeability describes how easily gas particles can pass through a given porous material. This is also dependent on the type of waste, in particular its density and settling properties. All of these parameters interact, resulting in an overall resistance to the flow of gas through the porous medium. For example, if the generation rate or size of the landfill increases, the suction level would also need to be increased to account for the additional amount of gas produced or additional distance that the gas is travelling. Or, if the permeability of the waste is lowered, then the well suction would also need to be increased. This principle is explained further via Darcy's Law in Chapter 2. The interaction of these parameters is part of what makes the code development challenging and lies at the heart of the code testing. This will be discussed in more detail in Chapters 2 – 4.

A realistic waste matrix is highly heterogeneous, since different waste types are added in different locations. The designers and operators will be aware of the expected differences and therefore can assign distinct effective permeability values in different azimuthal directions around the well. Since the purpose of the well is to create a radial flow of gas to the well pipe, the polar geometry is convenient, but does not imply that the landfill is necessarily circular. In light of the above, the cross-section is divided into several sectors. Each sector is assigned a

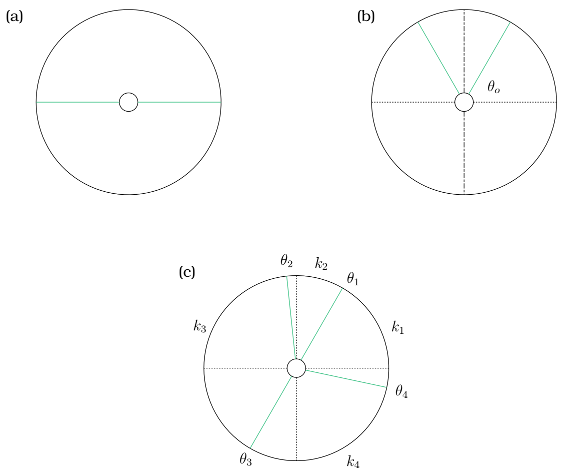


Figure 2: Sector configurations and counting notation of landfill well cross-sections for (a) semicircular sectors, (b) two unequal sectors and (c) arbitrary number of sectors (Nec and Huculak, 2020).

distinct permeability value. Mathematically it is required to number both the sectors and their boundary lines. Figure 2 shows the indexing of the rays / boundary lines is counterclockwise from the first non-negative angle. The sector indexing is also counterclockwise, starting below the first ray. For $s = N$ the index $s + 1$ refers to the next sector, which is $s = 1$.

As part of the testing procedure, these parameters will be divided into two main groups: anisotropy related parameters and generic physical parameters. The anisotropy group includes the number of sectors, the angles, and the permeabilities. From this point on, a given set thereof will be referred to as the landfill layout. The generic group pertains to any landfill, whether the modelling is isotropic or not, and includes the suction, outer radius, and generation rate. A fully defined setup of geometry and physical parameters constitutes a landfill configuration. The grouping of tests by layout and configuration is detailed in Chapter 2.

1.2 Analytical Solutions

The description below covers the principal equations of fluid flow through a porous medium, the conditions needed to obtain a valid flow solution in an anisotropic layout of the type shown in Figure 2, and the simpler instances of possible resultant flow fields (Nec and Huculak, 2020).

The equation of conservation of mass for all steady fluids as they flow through a porous medium is

$$\frac{1}{r}\frac{\partial}{\partial r}(r\rho u)+\frac{1}{r}\frac{\partial}{\partial \theta}(\rho v)=C, \tag{1.1a}$$

where r is the radius, θ represents the azimuthal angle, C is the generation rate, ρ is the density, u is the radial velocity component and v is the azimuthal velocity component. When considering the flow conditions of a landfill, it is reasonable to use the ideal gas equation equation, since both pressure and temperature are moderate (Young, 1989). Thus, the relationship between the density and pressure is given by $p=\rho RT$, where p, ρ, R, and T are pressure, density, ideal gas constant and temperature respectively (Young, 1989).

The balance of momentum equation is

$$\mathbf{u}=-\frac{k}{\mu}\nabla p=-\frac{k}{\mu}\left(\frac{\partial p}{\partial r},\quad \frac{1}{r}\frac{\partial p}{\partial \theta}\right), \tag{1.1b}$$

which describes how fluid moves through a granular bed. Equation 1.1b is also known as Darcy's law. Here μ is the fluid viscosity (in this case the fluid is landfill gas), k is the permeability of the medium (the waste) and the two components of $\mathbf{u}$ are u and v in Equation (1.1a). Equations (1.1a) and (1.1b) are combined with the ideal gas formula, giving

$$\frac{\partial}{\partial r}\left(k_s r\,\frac{\partial p_s^2}{\partial r}\right)+\frac{\partial}{\partial \theta}\left(\frac{k_s}{r}\,\frac{\partial p_s^2}{\partial \theta}\right)=-2\mu RTCr,\quad s=\{1,\,\ldots,\,N\}, \tag{1.2}$$

where the subscript $1\leq s\leq N$ denotes quantities that are distinct for different sectors.

$$\frac{\partial}{\partial r}\left(k_s r\,\frac{\partial p_s^2}{\partial r}\right)+\frac{\partial}{\partial \theta}\left(\frac{k_s}{r}\,\frac{\partial p_s^2}{\partial \theta}\right)=-2Cr,\quad s=\{1,\,\ldots,\,N\}, \tag{1.3a}$$

is Equation (1.2) in a dimensionless form, and

$$p_s^2(r,\theta_s)=p_{s+1}^2(r,\theta_s),\quad 1\leq s\leq N, \tag{1.3b}$$

and

$$k_s\,\frac{\partial p_s^2}{\partial \theta}\bigg|_{\theta_s^-}=k_{s+1}\,\frac{\partial p_{s+1}^2}{\partial \theta}\bigg|_{\theta_s^+},\quad 1\leq s\leq N. \tag{1.3c}$$

are the continuity conditions, which must hold on every contiguity ray θ_s in order for a solution to be valid. The boundary conditions for all of the solutions used in this project formally are

$$p\left(r_P,\theta\right)=p_P(\theta),\qquad p\left(r_X,\theta\right)=p_X(\theta) \tag{1.4}$$

where the well suction p_{out}, which is one of the main parameters discussed in the following sections is the mean of p_P. The mean of p_X is $0\,\text{inWC}$ (gauge pressure). The fluctuation in θ is small and is involved in the construction of the anisotropic solution (Nec and Huculak, 2020).

The simplest layout available is the semicircular one: the cross-section annulus is divided into equal bottom and top sectors with contiguity rays at $\theta_1 = 0$ and $\theta_2 = \pi$. Assigning respective permeabilities k_1 and k_2 to these sectors, the pressure in the top sector is given by,

$$P^{(2)}=-\frac{C}{2k_2}r^2+b_1+b_2\ln r, \tag{1.5a}$$

and similarly in the bottom sector,

$$P^{(1)}=\frac{C}{2}r^2\left\{\left(\frac{1}{k_1}-\frac{1}{k_2}\right)\cos(2\theta)-\frac{1}{k_1}\right\}+b_1+b_2\ln r. \tag{1.5b}$$

Note that the top sector solution is a purely radial function, whereas the bottom solution is angle dependent. The constants b_1 and b_2 are determined from boundary conditions. It is also possible to have

$$\begin{gathered}P^{(s)}=\frac{C}{2k_s}r^2\left(\frac{\cos(2\theta)}{\cos(2\theta_0)}-1\right)+b_1+b_2\ln r,\\ s=\{1,2\},\quad \theta_{s-1}\le\theta\le\theta_s,\end{gathered} \tag{1.6}$$

which is the equation for a landfill with a two sector configuration with boundary angles $\theta_1=\theta_0$ and $\theta_2=\pi-\theta_0$ with $0<\theta_0<\pi/2$ (Nec and Huculak, 2020). This equation will be explained further in Chapters 2 and 3.

For layouts with three and more sectors the most general solution used is:

$$P^{(s)}=\frac{C}{2k_s}r^2\left(a^{(s)}\sin(2\theta)+b^{(s)}\cos(2\theta)-1\right)+b_1+b_2\ln r,$$

$$\theta_{s-1} \leq \theta \leq \theta_s, \quad s = \{1, \ldots, N\}, \tag{1.7}$$

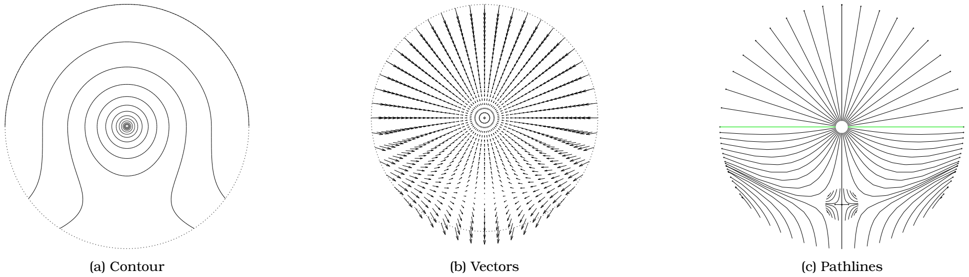
where the coefficients a and b satisfy a linear system of equations that must be solved numerically (Nec and Huculak, 2020). This will be explained further in Chapter 4.

1.3 Software

Due to the complexity of the analytical solutions, visualizing a landfill gas flow field requires software satisfying specific requirements. Two programs were used in this project, each with its own distinct and equally important purpose. The first program, VU, is predominantly a visualization software used to illustrate solutions of differential equations, enabling the display and manipulation of complex solutions for applications such as fluid dynamics, combustion, computational chemistry (VuChem). The second program, Octave, which is the open source equivalent of Mathworks' MATLAB, is a program designed for advanced numerical computations and is commonly used for problems including solutions of linear and nonlinear equations, statistics, linear algebra, and other numerical applications (Eaton et al., 2018). In this project, VU was mostly used in the initial stages to help with selecting visualization settings, which were then implemented in the Octave code. VU requires expertise in graphical visualization of vector fields to operate effectively and its interface is not easy to learn. As a result, it is rarely used outside academia. It is also limited as a computational tool. Therefore most of the project utilized Octave as the primary software. Once some visualization features were explored using VU, Octave was used to develop a more versatile code and visualize a wide range of landfill configurations. This will also be explained further in Chapter 2.

1.4 Visualization Methods of Landfill Gas Flow

There are three primary ways to visualize fluid flow solutions: pressure contours, velocity vectors and pathlines. Respective types of visuals are shown in Figure 3. Pressure contours are the curves of equal pressure in a given area, where the closer the lines are to one another, the steeper the pressure gradient (see Figure 3a). Due to the presence of very high gradients near the landfill centre, pressure contours do not provide enough detail. The contour representation provides the reader with information regarding the position in the landfill with the highest density of gas (simultaneously the highest pressure), from which one can deduce where the velocity would be the highest. However, in more complicated scenarios where one or more stagnation points are present, the contours alone do not provide the exact location of the stagnation points. In light of all the foregoing arguments, contours were eliminated as a visualization method prior to the beginning of this project.


(a) Contour

(b) Vectors

(c) Pathlines

Figure 3: Three visualization methods used to visualize fluid flow through a porous medium.

Vectors depict the direction and velocity of the gas, using the length of the arrows to represent the magnitude of the velocity (Figure 3b). VU and Octave were both used in the initial stages of developing the generic code. VU was used to make several decisions regarding basic visualization strategies. First, a mesh was created using Octave, as VU does not have the capability to create figures from equations alone. Next, a vector plot was created in VU. However, as shown in Figure 4, since the velocity of the landfill gas surrounding the well is orders of magnitude larger than at the perimeter of the landfill. This means that when the vector lengths are not artificially scaled, the arrows in the centre of the image entirely obstruct the rest of the figure, making the image unusable. When the length is scaled, an image of the type in Figure 3b ensues, but the information on the velocity magnitude is lost. Two more facts make this visual impracticable. One, observe that the flow field in Figure 3b contains a stagnation point. In its vicinity the velocities are even smaller than at the landfill perimeter, so much so that it is impossible to tell in what directions the arrow are pointing. Two, the well is an important part of the landfill, but its dimensions are small relative to the full landfill area. The natural arrangement in the polar coordinates is that near the well there will always be less space to draw the arrows than away from the well. Since over most of the domain the velocity magnitude trend is the opposite to what is conventional for vector representation, this type of visual simply cannot be improved.

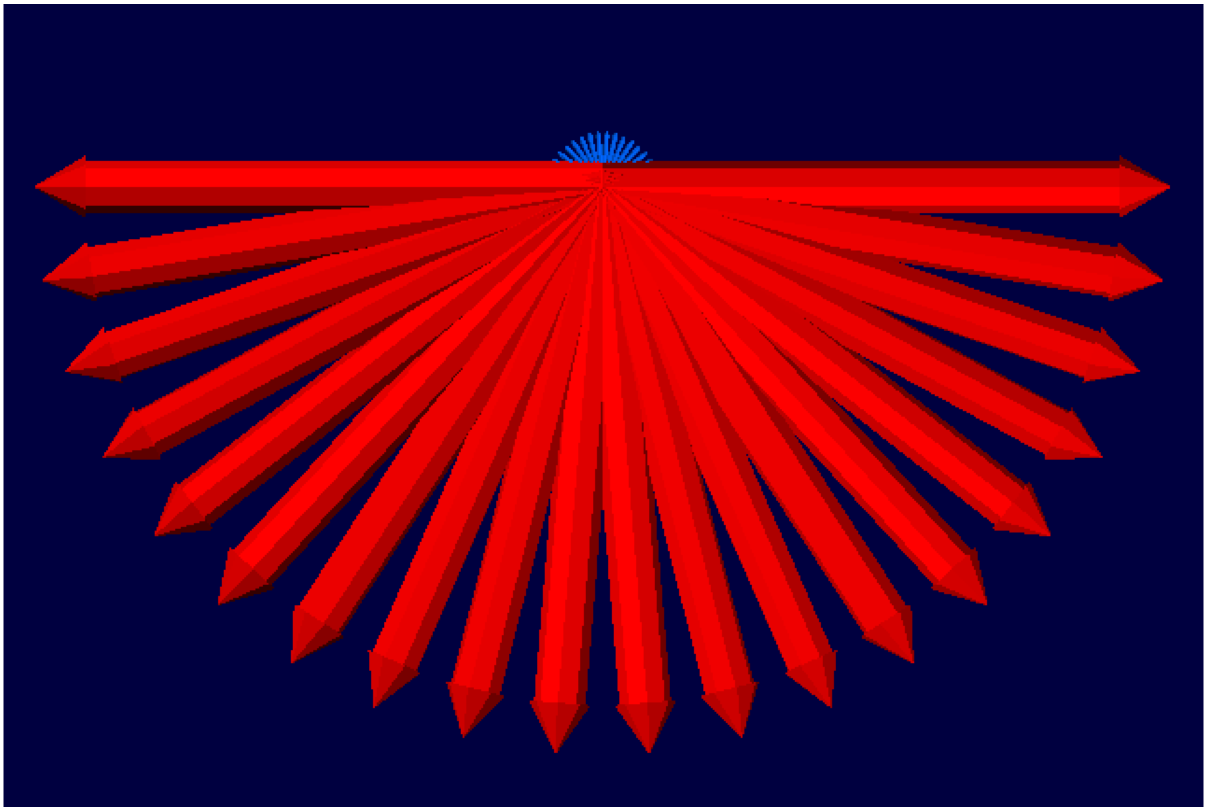
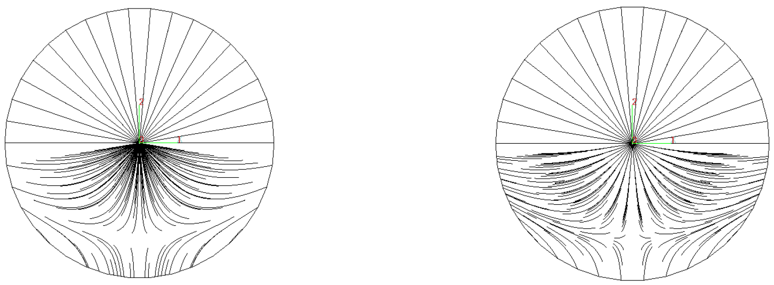


Figure 4: Vector representation created using VU, showing the issues occuring when the magnitude of the vectors is depicted to scale. Here the large red arrows represent gas flow with a high velocity immediately around the well, which completely obstruct the small blue arrows representing the gas flow with lower velocity around the perimeter of the landfill.

Finally, pathlines (Figure 3c) show the trajectory of a particle as it moves from a certain point. Pathlines were selected as the best visualization tool in this application, because they demonstrate the movement of the landfill gas and allow the reader to locate any stagnation points present within the area of the landfill. The visuals obtained with this method provide all information required by a landfill designer and do not involve an additional level of interpretation by the viewer as contours do. VU was used at the initial stage for obtaining both the vector and pathline representations.

Once the method of using pathlines was selected, the initial visualization setting was edited. This included using both forward and backward integration. A forward integration means the pathlines originate at the perimeter of the landfill and travel towards the well, just as would occur in reality. For backward integration the pathlines begin at the well or the close vicinity of a stagnation point and move in the direction opposite to the actual gas flow. Since the exact locations of both the well and the stagnation points are known, backward integration was selected to allow for each group of pathlines to originate at the region of highest importance as well as certainty (this will be briefly discussed further in §2.1.2). Forward integration also avoids unnecessary clustering of pathlines in the centre of the figure, which is an issue induced by the polar geometry, since the size of an element $r\mathrm{d}r\mathrm{d}\theta$ is not constant and quickly diminishes near the well. This problem is demonstrated in Figure 5.



(a) Forward integration

(b) Backward integration

Figure 5: Examples of forward and backward integration of pathlines for a landfill with two equal sectors, generated in VU.

1.5 Parameter Selection

To determine the robustness of the optimized code, many tests were run using a series of parameter combinations. Initially these variables included diameter of the pipe (d), well suction (p_{out}), landfill radius (r_x), generation rate (C), and the permeability of the top (k_t) and bottom (k_b) sectors. Hereinafter positive values of p_{out} correspond to negative gauge pressure relative to the atmospheric pressure (1 atm).

	p_{out}(inWC)	r_x(m)	C(scfm)	d(ft)
Minimum	5	5	50	0.25
Midpoint	25	25	500	0.5
Maximum	50	50	5000	1

Table 1: Parameter values for parametric scanning.

The method of parametric scanning was used to eliminate variables which did not have a significant effect on the output. Table 1 shows the combinations of high, low and midpoint values for four parameters, d, p_{out}, r_x, and C, that were run for a variety of different permeabilities. Based on preliminary tests, flow field patterns suggested a significant amount of repetition. In particular, three values of well diameter (d) were tested using the largest applicable range of the other parameters. During each of these tests, the three values of the well diameter were varied and the other variables were kept constant. Figure 6 is an example showing the similarity of the figures generated using the three values of the well radius, suggesting that the well diameter has little effect on the visualization and can be eliminated from future tests.

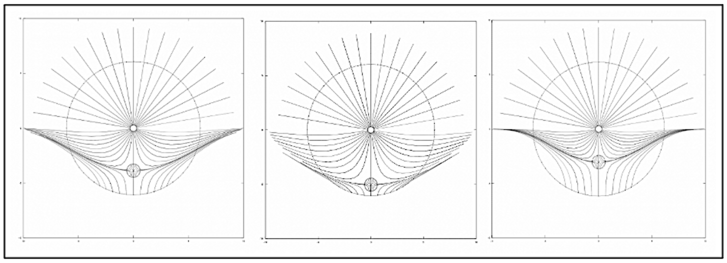


Figure 6: Three configurations with varied d values and all other parameters held constant.

The permeability values used when testing the code are listed below in Table 2 and were selected by considering the following. First, only orders of magnitude were tested, because resolution changes below this are too small and in practice it does not make sense to use more significant figures. Orders of magnitude for permeability are the standard for all porous media flows in large environmental applications such as flow of groundwater, shale gas, petroleum etc (Bear, 1972; Wikipedia, 2023). Second, the bottom sector always has a lower permeability than the top, however the layout can be rotated due to the rotational invariance of the equations, so the tests do cover the rare cases of a less dense bottom medium. Finally, the difference in orders of magnitude does not exceed three. In Chapters 3 and 4 only one order of magnitude difference was used.

Permeability of Bottom Sector $(\mathbf{m}^2)$	**Permeability of Top Sector** $(\mathbf{m}^2)$
$\mathbf{1e-05}$	$\mathbf{1e-04}$
$\mathbf{1e-06}$	$\mathbf{1e-05}$
$\mathbf{1e-07}$	$\mathbf{1e-06}$
$\mathbf{1e-08}$	$\mathbf{1e-07}$
$\mathbf{1e-09}$	$\mathbf{1e-08}$
$\mathbf{1e-10}$	$\mathbf{1e-09}$
$1e-06$	$1e-04$
$1e-07$	$1e-05$
$1e-08$	$1e-06$
$1e-09$	$1e-07$
$1e-10$	$1e-08$
$1e-07$	$1e-04$
$1e-08$	$1e-05$
$1e-09$	$1e-06$
$1e-10$	$1e-07$

Table 2: All possible permeability combinations used for the top and bottom sectors for two sector runs. Note: the bolded permeability values were used for all semicircular sector runs and the permeabilities not bolded were only used for the code developed from Equation (1.6).

Literature Review

Optimizing the visualization of gas flow in porous media is still a relatively unexplored area of applied mathematics. Because of its novelty the literature on the topic is limited. The few papers featured below encountered similar visualization difficulties in a range of applications, from the velocity of electro-osmotic flows through modelling groundwater to mapping the gas flow in drying mechanisms. Since the focus of those studies was on the physical problem itself, the quality of the visualization was not prioritized.

Most studies of gas diffusivity in soils assume isotropic media, however the article ’Gas diffusivity in soils compared to ideal isotropic porous media’ by Kuhne et al. (2012) looks at the modelling of gas diffusivity in anisotropic soil environments. There are many different physical measurements that can be made during an analysis of soil cores (root density, aeration porosity, and gas diffusivity), however 3D modelling is essential to integrate all possible diffusion pathways. While this paper includes modelling of gas diffusion within a porous medium (in this case, soil), the end goal for this paper and many others reviewed were graphs or figures demonstrating the observational results of their experiment rather than the visualization of the gas flow velocities (Kuhne et al., 2012).

In the paper 'Modelling Groundwater Flow Fields Containing Point Singularities: Streamlines, Travel Times and Breakthrough Curves', the modelling of steady state groundwater flow fields containing injection and extraction wells was explored. Figure 7 shows that the streamlines seem to originate at one point, which is the well. In reality, streamlines cannot intersect, because the path of the particle released at a given location is predetermined, while an intersection would mean that the particle can switch to a different path or that several particles might collide. In Figure 7, the streamlines are extremely dense near the well, creating an inaccurate representation (Charbeneau and Street, 1979). This example provides another example where the goal of a study was to understand a physical problem rather than on producing quality visuals.

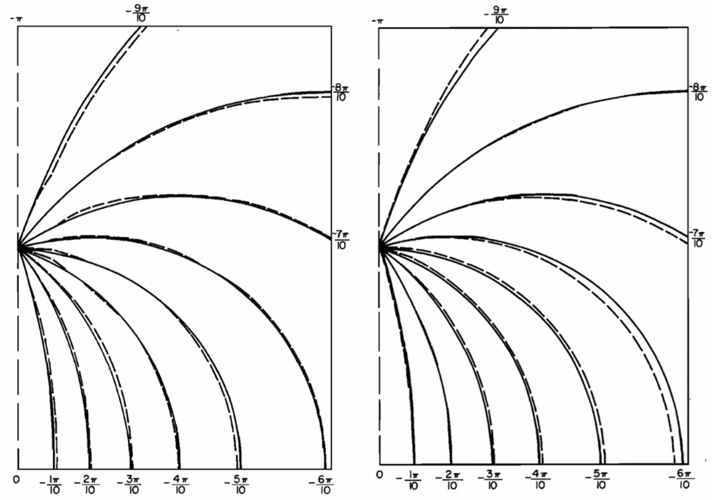


Figure 7: Analytical and numerically generated streamlines of the smoothed linear velocity field (left) and the quadratic velocity field (right) (Charbeneau and Street, 1979, Figure 3 and 4).

'Multi-functional Lagrangian flow structures in three-dimensional AC electro-osmotic micro flows', explores three-dimensional micro-flows using AC electro-osmosis (ACEO) and utilizes Lagrangian fluid trajectories to determine transport properties geometrically. Many of the figures in this article represent the different numerical simulations of ACEO flows for different geometries and boundary conditions. Figure 8 shows the planar and normal velocities in the

2D cross section of the flow field. Due to the nature of the flow field being visualized, there are various issues which are apparent in these visuals. In Image 8(a) the first issue is that the vector lengths are quite small, making the figure difficult to read. Second, the stagnation point is difficult to identify. In fact it is not clear how many such points there are. Third, all around the circumference it is impossible to tell whether there is no flow at all or just no normal flow. This is partly related to the issue with the vector lengths. If there is slip flow along the boundary, it is impossible to tell which way the arrows point and whether there are stagnation points on the boundary too. In Image 8(c) the first issue stems from the presence of the boundary layers resulting from the high gradient, and second, it is unclear whether the pointed segments are in reality smooth when the image is enlarged. Finally, in Image 8(b) , the contour resolution is low. This is a consequence of a high density at the “source" point, poor density elsewhere, implying a highly non-uniform gradient. These issues are similar to ones which were encountered during this project.

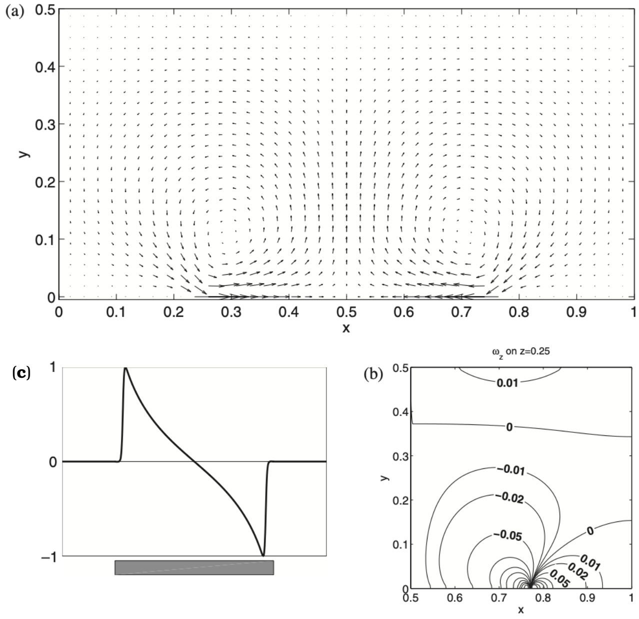
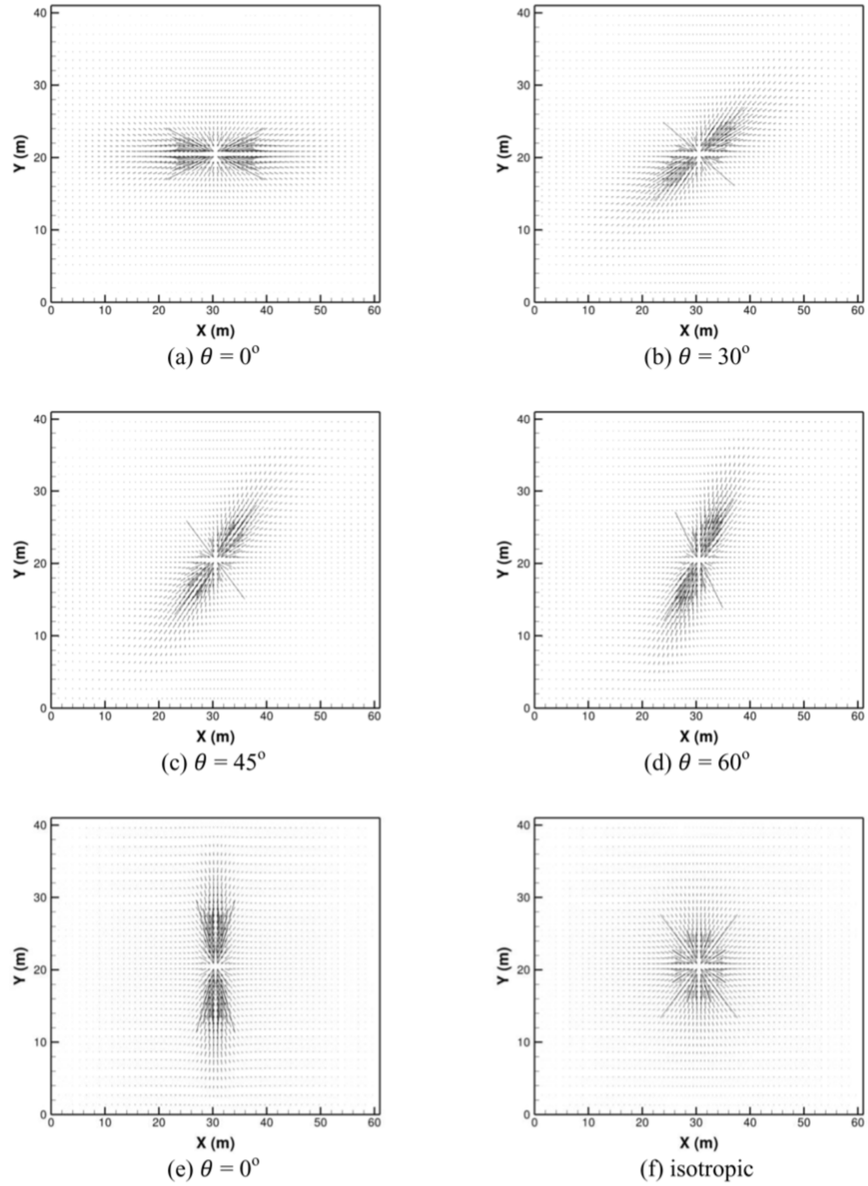


Figure 8: The planar velocity (a) and the normal velocity (b) of a two dimensional cross section of the flow field. Image (c) shows the slip velocity where the grey bar represents the electrode (Speetjens et al., 2011, Figure 4).

The article ’Numerical Simulation of Natural Gas Flow in Anisotropic Shale Reservoirs’ used a numerical model in order to understand how permeability, pressure gradients and anisotropy affect fluid flow in shale formations (Negara et al., 2015). All of the visualization issues are the same as those that were encountered in this project, however, it was not the goal of the article to optimize the visualization methods. Figure 9 shows an example using standard vector representation. It is evident that with the chosen figure, and the sizing of the arrows, it is difficult to understand what is happening with the flow field. It is also impossible to determine what type of stagnation point is present.

The foregoing observations regarding the priorities of each paper mentioned in the Literature Review are in no way a criticism of the work completed by each team. Instead, this review features an important gap in visualization methods, which supports the need for the tools developed in this project. When dealing with landfill gas solutions, one cannot use incomplete or poorly legible visuals, as it is important for landfill designers and operators to know how the gas is moving, where each stagnation point is located, and what kind of stagnation point it is. Access to such visuals thus provides additional tools to improve the effectiveness of the landfill. In other words, for the landfill industry the visuals are a working tool, not an end result.

It is also important to note that in the discipline of fluid dynamics colour is primarily used to depict an additional scalar field, separate from the primary entity being visualized. For example, colour may be used to show changes in temperature or pressure superimposed on a velocity field. Such double visualization is outside the scope of this project. Thus, all flow field diagrams will be displayed in greyscale. However, the visualization software that is based on the results of this project, and currently in evaluation by landfill industry professionals, does display the zone of collection in colour.



(a) $\theta = 0^{o}$

(b) $\theta = 30^{o}$

(c) $\theta = 45^{o}$

(d) $\theta = 60^{o}$

(e) $\theta = 0^{o}$

(f) isotropic

Figure 9: Standard vector representation of the velocity field distributions generated for the matrix and fracture (Negara et al., 2015).

2 Semicircular Sectors

The simplest landfill geometry was selected as a starting point for the development of the generic code. This geometry is a two sector semicircular layout where the contiguity rays are at $\theta_1 = 0$ and $\theta_2 = \pi$, and in practice each sector would represent a different type of waste or mixture of waste. There are several reasons behind starting with semicircular sectors (also referred to as two equal sectors). Keeping the angle of the boundary between the two sectors equal, there are fewer variables to account for. This means that when testing the robustness of the code (discussed in §2.2.1), far fewer runs are needed. Second, because of the axial symmetry inherent to the circle as a geometric shape, a two sector partition will always have a line of symmetry bisecting both. Thus it is possible to solve the problem on half the domain and simply reflect the pathlines to the other half. This saves computation time and simplifies testing. Third, from the point of view of the landfill designers and operators, having simple top versus bottom sector designations is more convenient to assess.

2.1 Methodology

There are two possible solutions for this layout, which are given by (1.5) and (1.6). For the majority of configurations explored in this chapter, the pathlines in the top sector are radial, since Equation 1.5a is not angle dependent. However, it is also possible to obtain the solution for two equal sectors from Equation (1.6), which is the solution for any two sector landfill geometry (Nec and Huculak, 2020). Since the functional shape is the same for the top and bottom sectors, Equation (1.6) produces pathlines which are angle dependent in both sectors.

2.1.1 Stagnation Points

Not all combinations of variables allow for an efficient landfill. A sufficient condition to deem a landfill poorly functional is the presence of one or more stagnation points. In fluid dynamics a stagnation point is the point in a flow field where all components of the velocity vector are zero. In this application a stagnation point can also be referred to as a flow reversal point, as this marks the location where the landfill gas is no longer being influenced by the well suction and therefore the gas beyond this point is flowing in the outward direction. Essentially, when a stagnation point is present in the flow field of a landfill, beyond that point the gas will not be collected by the well and therefore escape to the surrounding environment. Depending on the solution used, it is possible to have a variety of different stagnation point constellations and the number of stagnation points depends on the overall resistance acting on the gas. The possible locations were determined by taking the derivative of Solutions (1.5) or (1.6), first with respect to r and then with respect to θ and setting these derivatives to zero, to solve for the critical point. The derivative with respect to r is

$$\frac{-Cr}{k_2} + \frac{b_2}{r} = 0 \tag{2.1}$$

for Equation (1.5a). Since the top sector is radial, there is no θ component. The derivatives with respect to r and θ for Equation (1.5b) are

$$Cr\left\{\left(\frac{1}{k_1} - \frac{1}{k_2}\right)\cos(2\theta) - \frac{1}{k_1}\right\} + \frac{b_2}{r} = 0, \tag{2.2a}$$

and

$$\sin(2\theta) = 0 \tag{2.2b}$$

respectfully. Finally, the derivatives with respect to r and θ for Equation (1.6) are

$$\frac{Cr}{k_s}\left(\frac{\cos(2\theta)}{\cos(2\theta_0)} - 1\right) + \frac{b_2}{r} = 0, \tag{2.3a}$$

and

$$\sin(2\theta) = 0. \tag{2.3b}$$

Both (2.2b) and (2.3b) result in the same candidate angles: 0, $\pi/2$, π and $3\pi/2$. For each angle the radius is then calculated from either (2.2a) as

$$r^2 = -\frac{b_2}{C}\left\{\left(\frac{1}{k_1} - \frac{1}{k_2}\right)\cos(2\theta) - \frac{1}{k_1}\right\}^{-1} \tag{2.4a}$$

or (2.3a) as

$$r^2 = -\frac{b_2 k_s}{C}\left(\frac{\cos(2\theta)}{\cos(2\theta_0)} - 1\right)^{-1}, \tag{2.4b}$$

and a check is performed, to show that the radius is indeed less than the tested landfill radius. From the vantage point of physics there must be a stagnation point present at some point, which leaves the question of whether it is within the landfill or not. However, the solution with the radial top sector is almost unphysical in this regard, since it cannot have a realistic isolated stagnation point. Instead it has a whole arc of stagnation points, because the azimuthal velocity is 0 in the entire sector. This arc lies very very far out, where the radial velocity reverses. This is why the isotropic model is not reliable, as it predicts that the well drains an area orders of magnitude larger than in reality.

2.1.2 Code Development

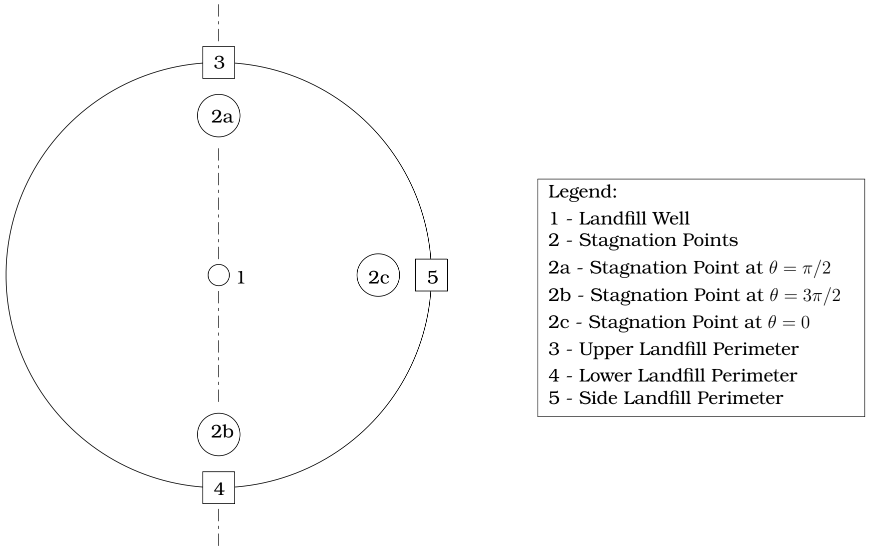
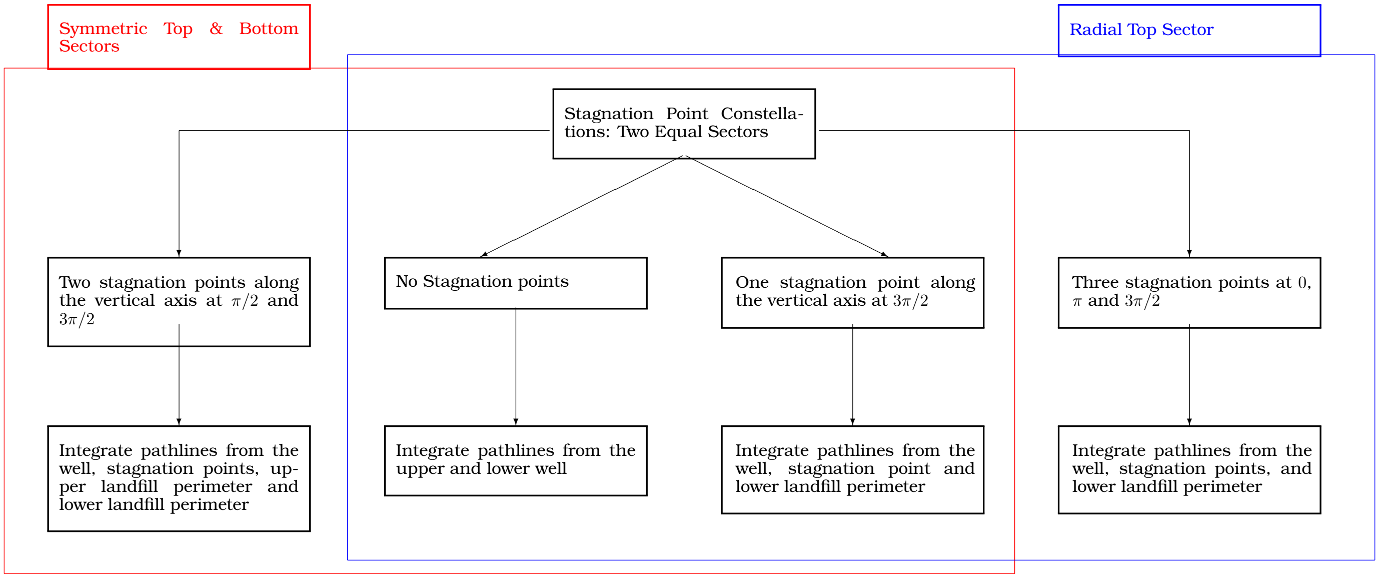

3
2a
1
2c
5
2b
4
Legend:
1 - Landfill Well
2 - Stagnation Points
2a - Stagnation Point at $\theta = \pi/2$
2b - Stagnation Point at $\theta = 3\pi/2$
2c - Stagnation Point at $\theta = 0$
3 - Upper Landfill Perimeter
4 - Lower Landfill Perimeter
5 - Side Landfill Perimeter

Figure 10: Regions of integration seeding where the numerical order corresponds to the frequency of occurance.

To achieve a reasonable density of pathlines throughout the domain, multiple points of interest were designated as origins of integration. These regions, shown in Figure 10, were selected based on the areas of the landfill, which provided the most important information, including a circle around the landfill well (1), circles around any stagnation points within the radius of the landfill (2) and any perimeter arc lying on the other side of a stagnation point (3-5). For landfills with two equal sectors, two solutions were used to integrate the pathlines: the first example, which will be referred to as Type I, occurs when the top sector has a radial boundary condition that results in purely radial flow in that sector (Equation (1.5)), and the second example, which will be referred to as Type II, is where both sectors have an azimuthal component (Equation (1.6)). For landfills with two equal sectors, where the flow in the top sector is radial (Type I), these regions or initial points of integration were designated based on reasonably uniform coverage of the landfill area by pathlines. However, for landfills with two equal sectors, where the top and bottom have symmetrical boundary conditions (Type II), or two unequal sectors, a hierarchy was found that made it possible to rank the regions by frequency of occurrence and integrate in that order. This hierarchy is also explained by Flowchart 1, which shows how the implemented code settings correspond to the features present in each landfill configuration. The simplest configuration, referred to as the basic group, does not have any stagnation points, therefore the only integration seeding necessary is from the well. All the particles follow a trajectory that is close to a straight line from the well to the perimeter of the landfill (showing that the gas should flow directly to the well in practice). The lower branches on the flow chart describe how the presence of one (occurring in both Type I and Type II configurations), two (in Type II configurations) or three (in Type I configurations) stagnation points affects the integration requirements. If any of the stagnation points are within the radius of the landfill, the pathlines are integrated from the circle around the well, a circle around each stagnation point and the corresponding landfill arc. The seeding circle around each stagnation point provides good coverage near the stagnation point and shows how the flow behaves in that region of slow motion. Since the stagnation point in essence forms a barrier, it is necessary to seed a new set of pathlines on the perimeter arc on its other side to cover the entire desired area. Finally, if the circle surrounding the stagnation point intersects with the circle around the well, pathlines from the well, which cross into the region around the stagnation point, are entirely removed. The code for pathlines in Type I configurations appears in Appendix A and pathlines in Type II configurations appears in Appendix B.

In all cases the function ode45, Octave's main built-in integration facility for ordinary differential equations, was used to convert the flow velocities, known in closed form from the analytical solutions, to particle positions within the circular domain. The seeding described above provides the initial positions from which the integration proceeds to create particle trajectories, i.e. pathlines.


Symmetric Top & Bottom Sectors
Radial Top Sector
Stagnation Point Constellations: Two Equal Sectors
Two stagnation points along the vertical axis at $\pi/2$ and $3\pi/2$
No Stagnation points
One stagnation point along the vertical axis at $3\pi/2$
Three stagnation points at 0, π and $3\pi/2$
Integrate pathlines from the well, stagnation points, upper landfill perimeter and lower landfill perimeter
Integrate pathlines from the upper and lower well
Integrate pathlines from the well, stagnation point and lower landfill perimeter
Integrate pathlines from the well, stagnation points, and lower landfill perimeter

Flowchart 1: Regions of integration according to number and location of stagnation point(s) for landfills with two equal sectors.

The length of integration is used to determine the length of the pathlines. Initially, each integration length was set manually for each configuration, which was essentially determined using trial and error: if the pathlines were too long or too short the length of integration for that region was changed. To remove unnecessary variables and therefore improve the versatility of the code, an integration halt was included. This mechanism made it possible to set an integration length exceeding any length required by the problem, and then have integration stop along any individual pathline until it reaches its desired location. For most lines this point is the perimeter of the landfill.

2.2 Results & Discussion

In order to have confidence in the robustness of the code being developed, it was crucial to learn the limitations of the physically feasible input. To achieve this, the variables listed in Tables 1 and 2 were combined to observe the maximal range of landfill scenarios. As mentioned previously , two separate codes for a landfill with two equal sectors were developed. Therefore throughout this section, the two versions will be described as either Type I or Type II, where Type I was developed from Equation 1.5 and Type II was developed from Equation 1.6.

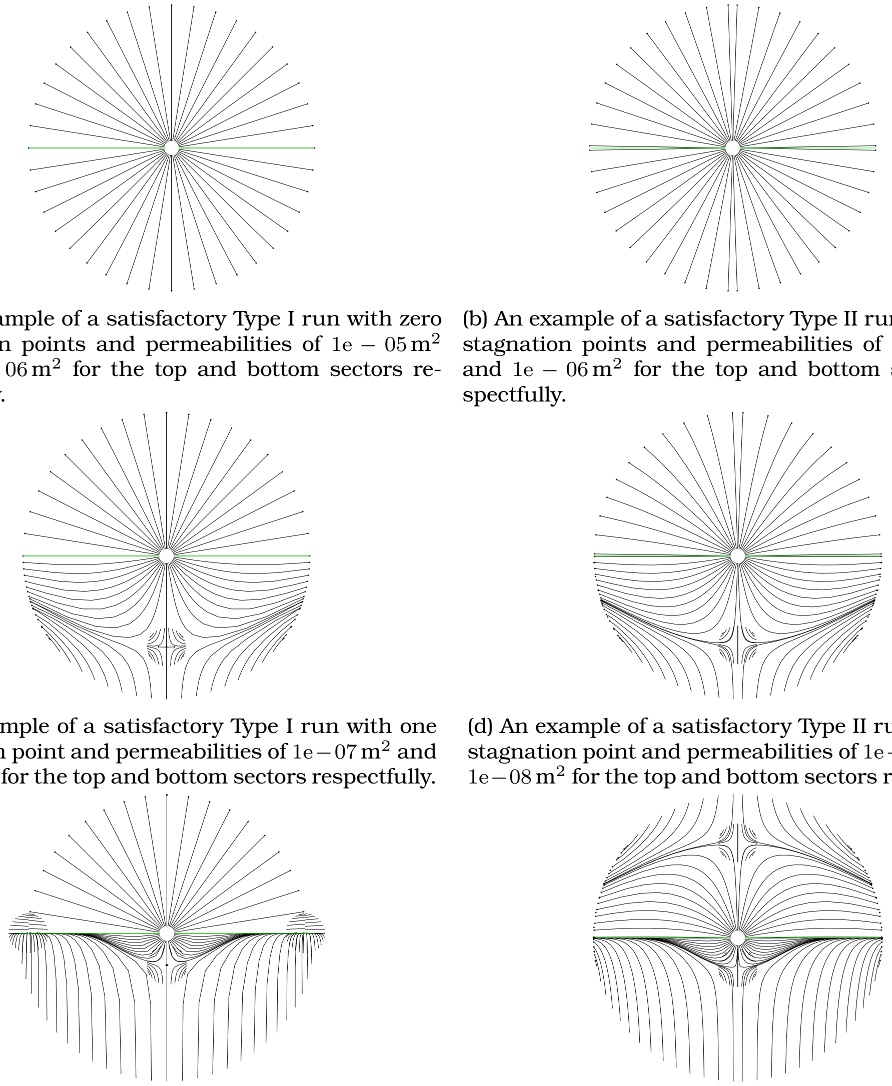
2.2.1 Classification of Run Success

Determining the robustness and versatility of the code is a critical step. This was done by running a series of parameter combinations using the values from Table 1 and a range of permeability arrangements shown in Table 2. To determine whether a code is robust, first the classification of what makes an integration satisfactory was defined. The success of a run was defined as: satisfactory (Y), unsatisfactory (N) or borderline (B). An example of this is shown in Table 3. All run classification data were stored in a spreadsheet file `ClassificationOfRunSuccessData` that accompanies the thesis and can be found at `https://tru.arcabc.ca/islandora/object/tru%3A6181` (Boham, 2023). The complete set of results for the current case is tabulated in the supplementary material in spreadsheets `2 Equal (Type I)` and `2 Equal (Type II)`. For Type I configurations a total of 405 tests were run resulting in 240 satisfactory, 23 borderline and 142 unsatisfactory configurations, and for Type II configurations 162 tests were run resulting in 112 satisfactory, 10 borderline and 13 unsatisfactory configurations. The classification was done by manual scrutiny of each image to determine whether the run fit the criteria for each of the mentioned classification groups. This process positively confirmed the code could handle a wide range of landfill configurations.

	p_{out} (inWc) 5, 25, 50	r_x (m) 5, 25, 50	C (scfm) 50, 500, 5000	Run Success Classification Satisfactory (Y), Borderline (B), Unsatisfactory (N)
1	5	5	50	Y
2	5	5	500	Y
3	5	5	5000	N
4	5	25	50	Y
5	5	25	500	Y
6	5	25	5000	N
7	5	50	50	Y
8	5	50	500	Y
9	5	50	5000	N
10	25	5	50	Y
11	25	5	500	Y
12	25	5	5000	Y
13	25	25	50	Y
14	25	25	500	Y
15	25	25	5000	Y
16	25	50	50	Y
17	25	50	500	Y
18	25	50	5000	Y
19	50	5	50	Y
20	50	5	500	Y
21	50	5	5000	Y
22	50	25	50	Y
23	50	25	500	Y
24	50	25	5000	Y
25	50	50	50	Y
26	50	50	500	Y
27	50	50	5000	Y

Table 3: Results of a code robustness test for landfills with two equal sectors (radial top sector) and permeabilities of $k = [1\mathrm{e}-07\,\mathrm{m}^2\ 1\mathrm{e}-06\,\mathrm{m}^2]$.

In a satisfactory run, the pathlines are relatively evenly spaced out, they cover the desired area of the landfill (regardless of the presence or absence of stagnation points) and the integration is halted at the correct location.



(a) An example of a satisfactory Type I run with zero stagnation points and permeabilities of $1\mathrm{e}-05\,\mathrm{m}^2$ and $1\mathrm{e}-06\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(b) An example of a satisfactory Type II run with zero stagnation points and permeabilities of $1\mathrm{e}-05\,\mathrm{m}^2$ and $1\mathrm{e}-06\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(c) An example of a satisfactory Type I run with one stagnation point and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(d) An example of a satisfactory Type II run with one stagnation point and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(e) An example of a satisfactory Type I run with three stagnation points and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

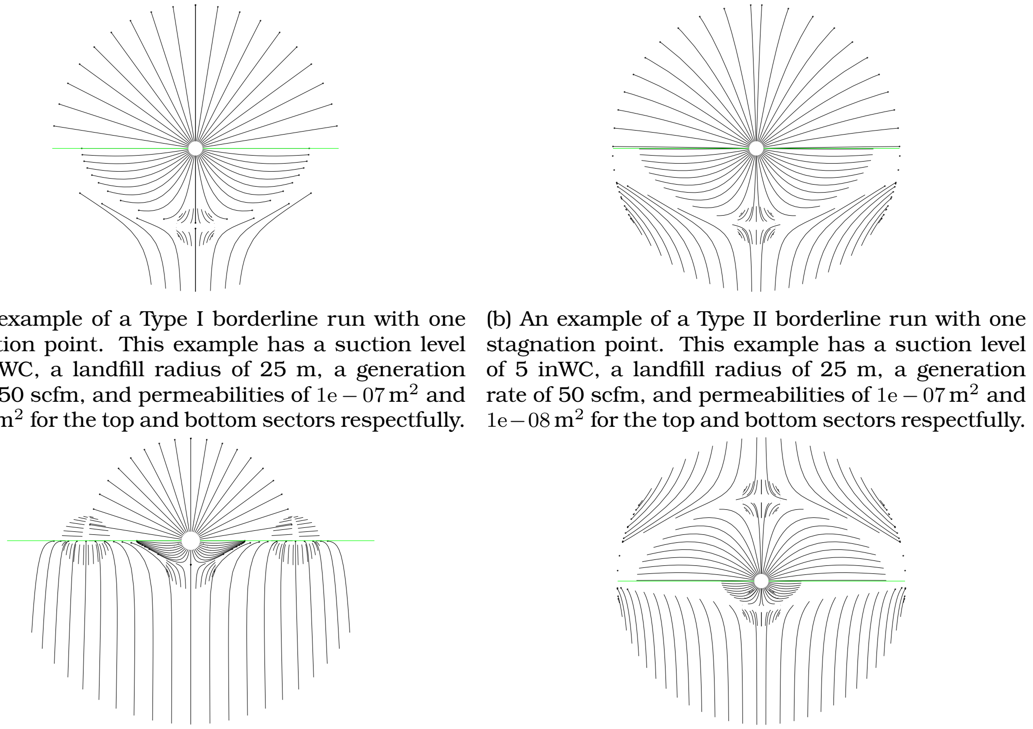
(f) An example of a satisfactory Type II run with two stagnation points and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

Figure 11: Comparison between satisfactory TypeI and Type II runs with varying number of stagnation points. All examples in this figure have a suction level of 5 inWC, a landfill radius of 5 m, and a generation rate of 50 scfm. The division between the two sectors is shown with the green line, idem throughout.

Figure 11 shows six different examples of satisfactory runs. Subfigures 11a, 11c and 11e, show configurations where the top sector is radial (Type I) and the three images on the right (Subfigures 11b, 11d and 11f), show configurations where both the top and bottom sectors are θ dependent (Type II). In Subfigures 11a and 11b the permeabilities are high, meaning there are no stagnation points present and therefore all the pathlines (landfill gas) reach the landfill well. These two figures also do not differ all that much, just the slightly curved top pathlines, so in reality the debate over which solution is better is often moot. n Subfigures 11c and 11d the permeabilities are moderate, which results in one stagnation point. This means that some pathlines do not reach the landfill well and suggests that the landfill configuration would result in gas escaping the collection system. In Subfigures 11e and 11f the permeabilities are low, resulting in either three stagnation points (in systems where the top sectors are radial) or two stagnation points (where both sectors are dependent on the angle of the starting point of the pathlines). This means that even more of the gas is escaping the system. In Subfigure 11e note that the two stagnation points along the horizontal axis are really only half stagnation points. This peculiar status is due to the anisotropy, since the medium permeability jumps abruptly across the contiguity ray, whilst the stagnation point is directly on the ray. Therefore when the points happen to be on the ray, it is harder to predict what is going on in terms of the flow field, since such critical points are not conventional. Subfigures 11e and 11f are also examples of scenarios where a stagnation point circle encroaches on the well circle area and the interference had to be resolved by the addition of an appropriate integration halt. Finally, despite the fact all the images in this figure are classified as successful, in practice the only two which represent a properly functional landfill set up are Subfigures 11a and 11b. Subfigures 11c and 11d are fine but still show some escaping gas therefore this would tell the engineers to instal an additional well at the stagnation point location.

A borderline visual is partially balanced (may include minor coverage gaps) and the pathlines do not all reach the perimeter of the landfill. Some examples are given in Figure 9. These visuals provide some information about the movement of the gas, however it is clear that the visual is lacking if one were to use it for the purpose of flow field interpretation: due to the properties of the landfill and the settings of the code, some of the lines are not fully integrated. It would be possible to obtain figures where the pathlines are fully integrated, provided a custom integration time was set. In practice, if a visual cannot be obtained within a reasonable period of time, it is a clear indication to the designer or operator that the input configuration is not an effectively functioning landfill. A reasonable visual construction time was determined in consultation with landfill engineers (GNH Consulting Ltd., Delta, British Columbia) to be

60sec or less on a home computer of moderate technical specifications. The borderline visuals are sufficient to infer which input parameters require modification.



(a) An example of a Type I borderline run with one stagnation point. This example has a suction level of 5 inWC, a landfill radius of 25 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(b) An example of a Type II borderline run with one stagnation point. This example has a suction level of 5 inWC, a landfill radius of 25 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

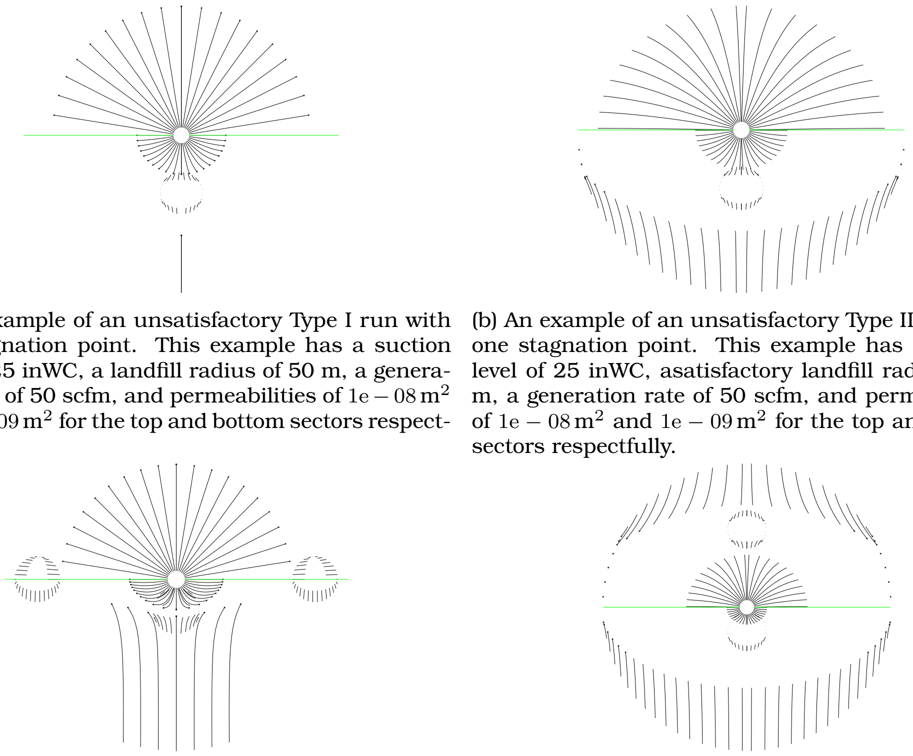
(c) An example of a Type I borderline run with three stagnation points. This example has a suction level of 25 inWC, a landfill radius of 50 m, a generation rate of 500 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(d) An example of a Type II borderline run with two stagnation points. This example has a suction level of 5 inWC, a landfill radius of 25 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

Figure 12: Comparison between Type I and Type II borderline runs with one or two stagnation points.

Subfigures 12a and 12b show examples of Type I and II landfill configurations respectively with one stagnation point. Subfigure 12c shows a landfill configuration with three stagnation points. Like Subfigure 12a, this is also an example of a Type I configuration. This is a unique arrangement where two stagnation points lie on the contiguity ray, and as the image shows, only half the pathlines behave as expected near a typical saddle point. Subfigure 12d shows another Type II landfill configuration with two stagnation points. All of these examples provide the necessary information regarding how the gas is behaving without the pathlines filling the complete landfill area. For all these examples, the order of integration follows Flowchart 1.

An unsatisfactory run is one that has major structural issues and could not be accepted as borderline. For example, issues such as overlapping pathlines that originated from different integration regions (this is an indication of the fact the integration was not accurate enough, since in a correct flow field pathlines cannot intersect) or major gaps in coverage. Figure 13 shows four different examples of unsatisfactory runs, where crucial information that the figure should provide, is missing.



(a) An example of an unsatisfactory Type I run with one stagnation point. This example has a suction level of 25 inWC, a landfill radius of 50 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(b) An example of an unsatisfactory Type II run with one stagnation point. This example has a suction level of 25 inWC, asatisfactory landfill radius of 50 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(c) An example of an unsatisfactory Type I run with three stagnation points. This example has a suction level of 5 inWC, a landfill radius of 25 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

(d) An example of an unsatisfactory Type II run with two stagnation points. This example has a suction level of 5 inWC, a landfill radius of 50 m, a generation rate of 50 scfm, and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectfully.

Figure 13: Comparison between unsatisfactory Type I and Type II runs with one, two, or three stagnation points.

Subfigure 13a shows a configuration with one stagnation point. This configuration is an example which was also generated from Equation 1.5, and therefore has a radial top sector. Subfigure 13b also shows a landfill configuration with one stagnation point. This configuration was generated from Equation 1.6, and all the pathlines are dependent on the angle. The example in subfigure 13c shows an unsatisfactory configuration with three stagnation points. Like Subfigure 13a, Subfigure 13c was generated from Equation 1.5, which means the pathlines in top sector are also radial. Subfigure 13d gives an example with two stagnation points and was was generated from Equation 1.6, therefore just like Subfigure 13b all the pathlines are angle dependent.

In the code developed from Equation (1.6), there were also instances where regardless of the settings the integration would not complete. Such scenarios were designated by an X in our evaluation of run success.

Finally it is important to note that regardless of whether a run is deemed satisfactory or not, the classifications previously discussed were entirely based on the ability to produce an image. This does not determine whether the landfill itself would function efficiently. In a correct design, none of the generated gas should be escaping from the well, therefore the presence of one or more stagnation points would imply that those landfill parameters were not ideal.

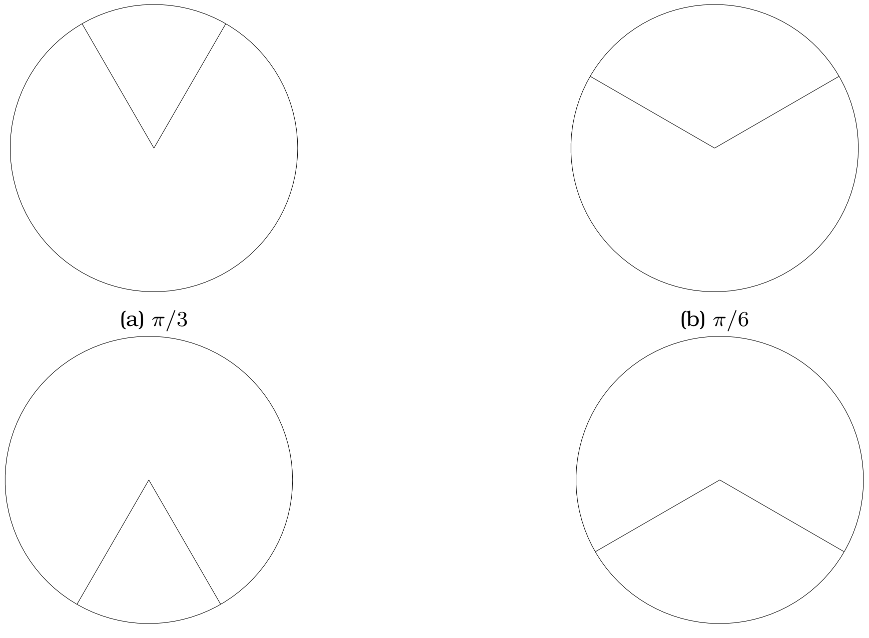
3 Two Sectors of Unequal Angles

The next logical step in code development and analysis was a landfill with two sectors of unequal angles. Including the boundary angle between sectors as an additional variable increased the complexity of the flow field and thence the choices made in the code in order to create effective visuals. Many observations and code development methods determined in Chapter 2 were the backbone for approaching the visualization of a landfill configuration with two unequal sectors, and the differences are detailed in this chapter.

3.1 Parameter Selection & Code Development

The values of parameters selected for two unequal sectors were predominantly the same as for two equal sectors. These include suction applied to the well, landfill radius, generation rate, and the permeability of the top and bottom sectors.Unlike the case of two equal sectors, the angle between the two sectors is no longer constant, therefore when testing the robustness of the generic code, various sector angles were tested. These are shown in Figure 14: $\pi/3$ and $\pi/6$ were selected as the angles used in the test runs as they fall on either side of $\pi/4$, which is a value where no solution exists and a cross-over occurs from stagnation points on horizontal to vertical or vice versa. Equation (1.6) confirms that it is not possible to have a boundary at $\pi/4$, since $\theta_0 = \pi/4$ would give a fraction with a zero denominator. This also explains why stagnation points, shown later in §3.2, jump between the horizontal and vertical axes. Since no steady state exists at $\pi/4$, it it not possible to locate the position of any stagnation points, and when the boundary passes through this angle, the flow field "jumps". This effect will be visualized below.

To ensure that all possible geometry settings were explored, the negative values for each of the two angles were selected. This also achieves the same outcome as swapping the permeabilities from the top to bottom sector and therefore mimicking the effect of the lower sectors having higher permeabilities. Overall these choices create a uniform coverage of the circular domain. The correspondence of all equations to a naturally occurring phenomenon and continuity in all parameters together ensure that any configuration lying between successfully tested ones would also result in satisfactory visuals.



(a) $\pi/3$

(b) $\pi/6$

(c) $-\pi/3$

(d) $-\pi/6$

Figure 14: Angles used in the classification of run success.

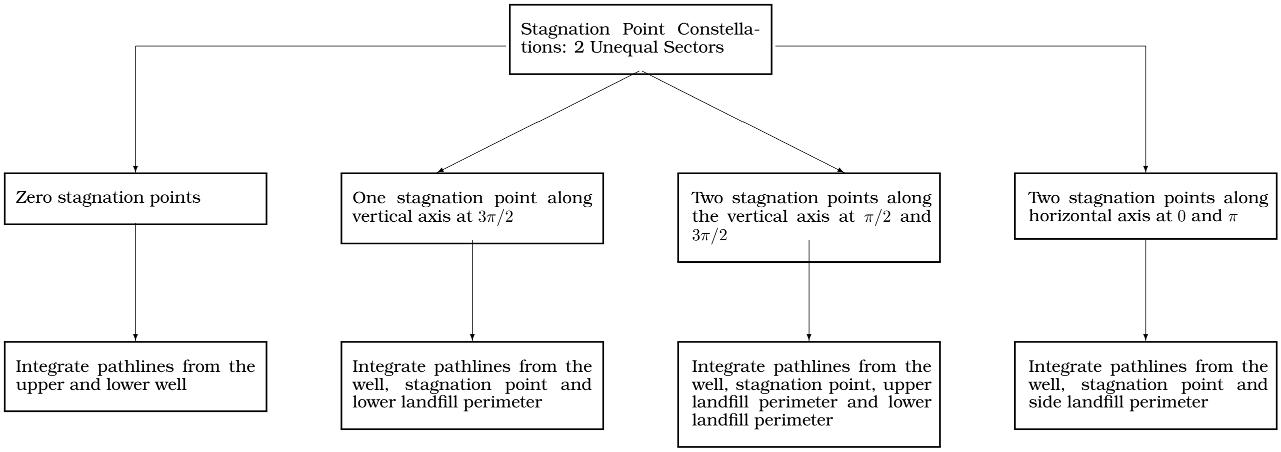
Since the landfill examples in this chapter are more complex than those in Chapter 2, the need for a second integration halt arose. This halt was added slightly outside of the original in an attempt to stop some of the runaway pathlines which persistently appeared. Despite this effort, some particularly complicated cases (runs with high resistance acting on the gas) had pathlines breaching the second halt. This is exemplified later in Figure 20. The halt execution during an integration is a built-in option in the Octave function `ode45`. The function checks whether the requested value was crossed and if yes, stops the integration. However, one limitation of `ode45` is that if the very first step overshoots the halt value, the halt procedure irrevocably fails. Thus adding an additional halt as an attempt to define one more threshold that the first step would not cross, is pointless, since the pathlines go too far for the visual to depict the landfill as desired.

3.1.1 Stagnation Points

Just as in the two equal sectors scenario, discussed in §2.1.1, the presence of any stagnation points shows the location where the landfill gas is no longer being influenced by the well's suction. For landfill configurations with low or moderate resistance, either zero or one stagnation point will be present at $3\pi/2$. For configurations with high resistance, two points are present, with their location is dependent on the angle of the boundary between the two sectors. When the angle is less than $\pi/4$, the two stagnation points are on the vertical axis at $\pi/2$ and $3\pi/2$, and when the angle is greater than $\pi/4$, the points are on the horizontal axis at 0 and π. The locations of these stagnation points were determined by differentiating Equation (1.6) with respect to r and θ and setting the derives to zero. Therefore Equations (2.3), are also used to solve for the critical points.

3.1.2 Code Development

The code for a landfill with two unequal sectors uses a hierarchy system of variables in order to determine which regions of the landfill need to be integrated. As shown in Figure 10, these regions are numbered according to the frequency of their occurrence. The integration around the landfill well is designated as region 1, as it is present in every configuration regardless of the number of stagnation points. The stagnation points are designated as region 2 along with the letters (a) through (c) according to increasing θ value. The letter designations are the only features that are not frequency dependent and are simply dependent on the increasing value of the angle θ_{sp}, at which the stagnation point is situated. Since the integration is symmetrical across the vertical plane, a point at π is not necessary as it is a reflection of the point at 0. Since a single stagnation point at $3\pi/2$ is the most common constellation of stagnation point, the lower landfill perimeter region is designated as region 3. The upper landfill perimeter is designated as region 4 and finally the side landfill perimeter is designated as region 5. All of these scenarios are summarized in Flowchart 2, where the regions or integration are entirely dependent on the number and location of the stagnation points.


Stagnation Point Constellations: 2 Unequal Sectors
Zero stagnation points
One stagnation point along vertical axis at $3\pi/2$
Two stagnation points along the vertical axis at $\pi/2$ and $3\pi/2$
Two stagnation points along horizontal axis at 0 and π
Integrate pathlines from the upper and lower well
Integrate pathlines from the well, stagnation point and lower landfill perimeter
Integrate pathlines from the well, stagnation point, upper landfill perimeter and lower landfill perimeter
Integrate pathlines from the well, stagnation point and side landfill perimeter

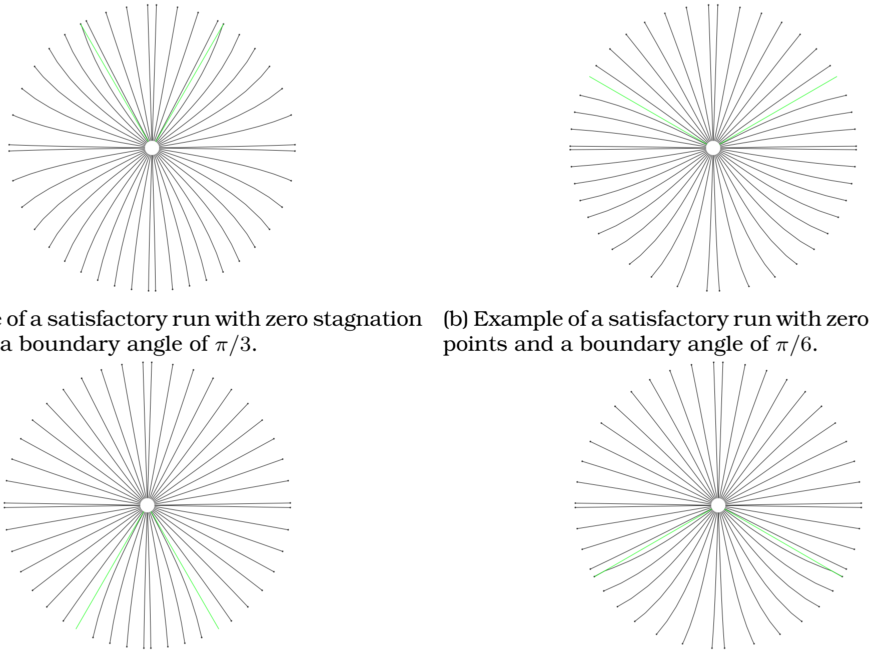
Flowchart 2: Regions of integration according to number and location of stagnation point(s) for landfills with two unequal sectors.

3.2 Results & Discussion

The testing of code robustness is performed in a similar manner to that described in §2.2.1, however, for two unequal sectors the additional variable of the sector angle increases the number of runs required. For this system, all of the variables listed in Tables 1 (excluding the diameter of the well) and the bolded permeabilities in Table 2 were combined with the values in Table 14. Overall, for the two unequal sectors system, 648 tests were run.

3.2.1 Classification of Run Success

The figures generated during the robustness tests were divided into the same four categories as in Chapter 2: satisfactory (Y), borderline (B), unsatisfactory (N), and unfinished (X). These results are tabulated in the spreadsheet file `ClassificationOfRunSuccessData` (Boham, 2023). The complete set of results for this Chapter is tabulated in the spreadsheet, `2 Unequal`. In sum, there were 411 satisfactory, 32 borderline, 45 unsatisfactory, and 160 failed configurations. Similar to Chapter 2, satisfactory runs are those which include all necessary details in order to be able to understand how the landfill gas is behaving in a particular system. All of the pathlines are proportionally spaced and reach the outer boundary without exceeding the landfill radius. Figure 15 includes four examples of satisfactory runs without stagnation points. Parameter values for these examples are on the higher side of low resistance, enabling the visualization of the differences between the different boundary angles. As the pathlines approach an area where a stagnation point is situated, the lines begin to bend. In Subfigures 15a the pathlines bend away from the horizontal line suggesting the presence of two stagnation points further along the horizontal axis. Subfigures 15b and 15d both show examples where the lines are bending outwards at the bottom of the landfill, implying that one stagnation point will appear within that sector before a second point is present elsewhere. This will be more apparent below when Figure 17 is discussed. Finally, Subfigure 15c follows the same trend as Subfigure 15a, though the pathline behaviour is more subtle.



(a) Example of a satisfactory run with zero stagnation points and a boundary angle of $\pi/3$.

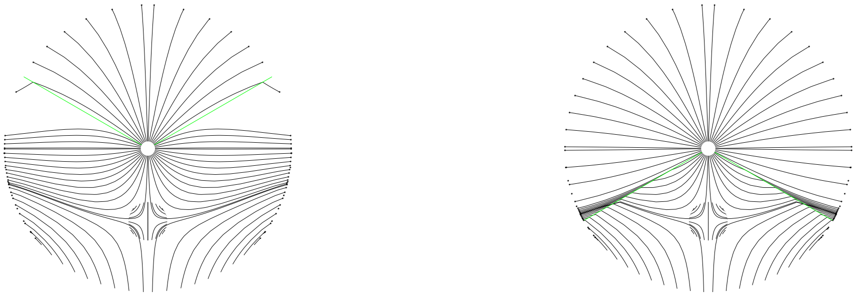
(b) Example of a satisfactory run with zero stagnation points and a boundary angle of $\pi/6$.

(c) Example of a satisfactory run with zero stagnation points and a boundary angle of $-\pi/3$.

(d) Example of a satisfactory run with zero stagnation points and a boundary angle of $-\pi/6$.

Figure 15: Comparison of satisfactory runs without stagnation points. The four landfill examples have a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectively.

Figure 16 includes two examples of satisfactory runs with one stagnation point. Such flow fields are only possible when the boundary angles are either $\pi/6$ (Subfigure 16a) or $-\pi/6$ (Subfigure 16b), because when the boundary angle is below $\pi/4$, the stagnation points lie along the vertical axis, i.e. each point is situated in a different sector and thus subjected to a different permeability. Therefore points on the vertical axis are not at an equal distance from the well, meaning it is possible for the resistance acting on the landfill gas to be just enough that only one point is present. As mentioned before, this will be more apparent when Figure 17 is discussed. Finally, the examples in Figure 16 are the first where breaking of pathlines as they cross the contiguity rays is observed. This phenomenon was also observed in an experimental application (Sahu and Flynn, 2015). The experimental set up involved mapping fluid flow through glass beads to demonstrate the break in flow field slope as the fluid moves from one permeability region to another (Sahu and Flynn, 2015).

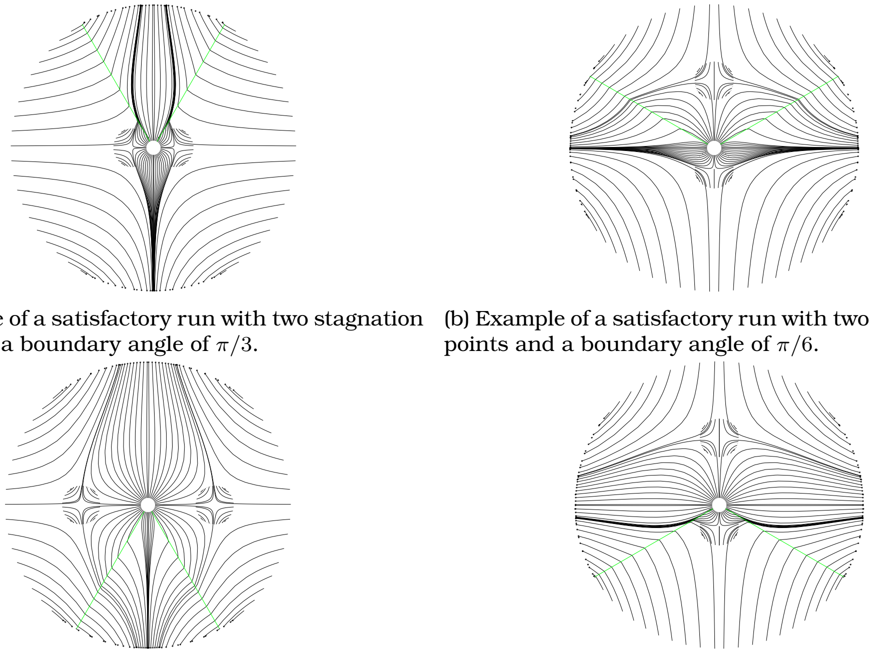


(a) Example of a satisfactory run with one stagnation point and a boundary angle of $\pi/6$.

(b) Example of a satisfactory run with one stagnation point and a boundary angle of $-\pi/6$.

Figure 16: Comparison of satisfactory runs with one stagnation point. The two examples have a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectively.

Figure 17 includes four examples of satisfactory runs with two stagnation points. Subfigure 17a shows an example where the boundary between the two sectors has an angle of $\pi/3$. Here the two stagnation points are situated along the horizontal axis very close to the landfill well, and their distances therefrom are equal. When the boundary angle is above $\pi/4$, the stagnation points lie on the horizontal axis, so the two points are positioned in the same sector. This means that there is the same resistance acting on the landfill gas, which results in the stagnation points being positioned at an equal distance from the well. The vertical symmetry is another reason for the equal distancing. Subfigure 17b shows an example where the boundary angle is $\pi/6$. Here the two stagnation points occur along the vertical axis however, with the lower point closer to the well since the bottom permeability is lower. Subfigure 17c shows an example where the boundary angle is $-\pi/3$. The two stagnation points here are also along the horizontal axis and their distances from the landfill well are equal just as in Subfigure 17a. However, despite the parameter values being the same for all examples in the figure, the stagnation points in Subfigure 17a are much closer to the well than Subfigure 17c because the size of the upper and lower sectors varies between the positive and negative versions of the angles, whereas the respective sector permeabilities remain the same. Subfigure 17d shows an example where the boundary angle is $-\pi/6$. Just as in Subfigure 17b, both stagnation points are along the vertical axis and the distances between each point and the well are not equal. Finally, all of the examples in Figure 17 show the pathlines breaking as they pass over the boundary.



(a) Example of a satisfactory run with two stagnation points and a boundary angle of $\pi/3$.

(b) Example of a satisfactory run with two stagnation points and a boundary angle of $\pi/6$.

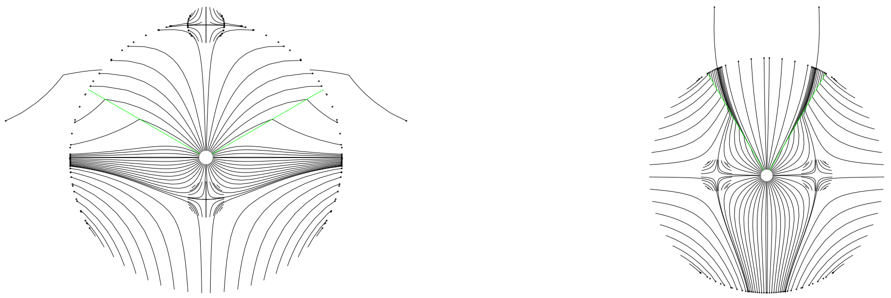
(c) Example of a satisfactory run with two stagnation points and a boundary angle of $-\pi/3$.

(d) Example of a satisfactory run with two stagnation points and a boundary angle of $-\pi/6$.

Figure 17: Comparison of satisfactory runs with two stagnation points. The four landfill examples have a suction level of 5 inWC, a landfill diameter of 25 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ for the top and bottom sectors respectively.

As discussed previously in Chapter 2, despite figures being classified as successful, only Figure 15 represents a functional landfill, as there are no stagnation points present and all pathlines are shown travelling towards the well.

Runs were classified as borderline for two primary reasons: first, if there were one or two lines which exceeded the perimeter of the landfill, and second, if there were minor gaps in the pathline density which did not affect the ability to determine what was occurring in the landfill. As mentioned in §3.1, a second integration halt was added for the purpose of preventing pathlines in some of the more difficult cases from escaping the landfill perimeter, and Figure 18 shows two examples of minor runaway lines. In the following paragraphs an analogue clock face will be used to describe the locations of runaway lines or gaps in integration. In Subfigure 18a the boundary angle was $\pi/6$, therefore the stagnation points are on the vertical axis, however the pathlines run away in the horizontal direction (from two and ten o'clock and in the direction of three and nine o'clock). In Subfigure 18b the angle of the boundary was $\pi/3$, which means both stagnation points are on the horizontal axis, however the pathlines runaway in the vertical direction (from one and eleven o'clock). The positioning of the stagnation points on the vertical means that the paths would have to bend when coming near. It is known that all points in two sector configurations are of the saddle type (Nec, 2021). In combination with vertical symmetry, this implies the bending has to be in the form of a "plus", so vertical lines become horizontal. Technically this means that we misjudged the seeding somewhat: the lines that look good actually came from the well. The points on the perimeter are the successfully halted lines and those two just did not halt, essentially the `ode45` step overshoots the set halt threshold.

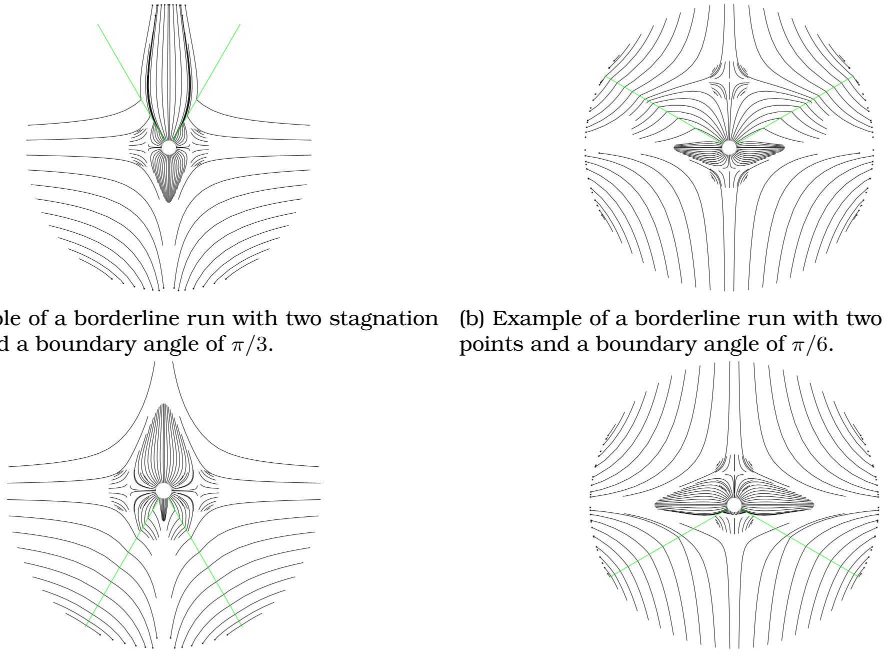


(a) Example of a borderline run with two stagnation points. The parameters include a well suction of 25 inWC, a landfill radius of 25 m, a generation rate or 5000 scfm, permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$ for the top and bottom sectors respectively, and a boundary angle of $\pi/6$.

(b) Borderline run with two stagnation points. The parameters include a well suction of 50 inWC, a landfill radius of 25 m, a generation rate or 5000 scfm, permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$ for the top and bottom sectors respectively, and a boundary angle of $\pi/3$.

Figure 18: Comparison of borderline runs which involve singular runaway lines.

Figure 19 gives examples of the second type of borderline visuals. Subfigure 19a shows an example where the boundary is $\pi/3$. Here, the gaps due to pathline length are along the vertical axis (specifically at six o'clock) which, just like the runaway lines, is perpendicular to the axis the stagnation points are positioned on. The boundary in Subfigure 19b is $\pi/6$, and demonstrates gaps in integration lying on the horizontal axis (directly between three and nine o'clock). Subfigures 19c and 19d are for the negative angles of $\pi/3$ and $\pi/6$ respectively. These figures follow trends similar to their positive counterparts: the gaps lie on the vertical axis (six o'clock) for $-\pi/3$ and on the horizontal axis (between three and nine o'clock) for $-\pi/6$. Finally, one interesting feature that is present in Subfigures 19a and 19c is the gap at the top left and right corners.



(a) Example of a borderline run with two stagnation points and a boundary angle of $\pi/3$.

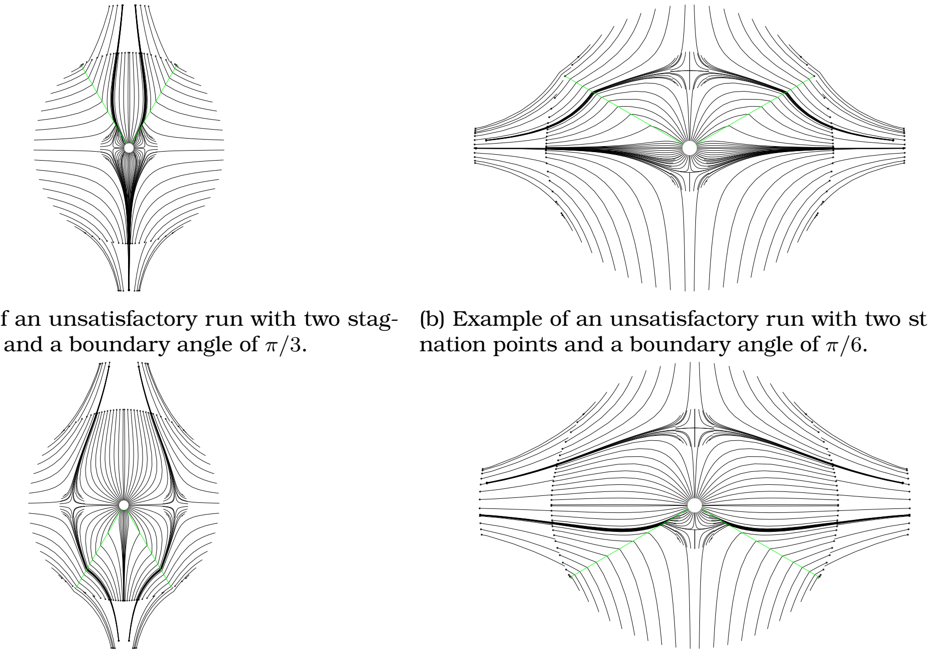
(b) Example of a borderline run with two stagnation points and a boundary angle of $\pi/6$.

(c) Example of a borderline run with two stagnation points and a boundary angle of $-\pi/3$.

(d) Example of a borderline run with two stagnation points and a boundary angle of $-\pi/6$.

Figure 19: Comparison of borderline runs which include moderate gaps within the figure. The four landfill examples have a suction level of 5 inWC, a landfill diameter of 25 m, and a generation rate of 50 scfm. The permeabilities of Subfigures 19a and 19b are $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectively. The permeabilities of Subfigures 19c and 19d are $1\mathrm{e}-09\,\mathrm{m}^2$ and $1\mathrm{e}-10\,\mathrm{m}^2$ for the top and bottom sectors respectively.

Finally, runs were classified as unsatisfactory if significant structural issues were present. These primarily present as examples with major gaps in the integration or examples with a significant number of pathlines escaping the integration halt. Figure 20 shows four examples of unsatisfactory runs with large numbers of runaway pathlines. These examples essentially show more extreme scenarios of the examples in Figure 18, where Subfigures 20a and 20c have stagnation points along the horizontal axis and runaway pathlines travelling along the vertical axis, and 20b and 20d have stagnation points along the vertical axis and runaway pathlines travelling along the horizontal axis.



(a) Example of an unsatisfactory run with two stagnation points and a boundary angle of $\pi/3$.

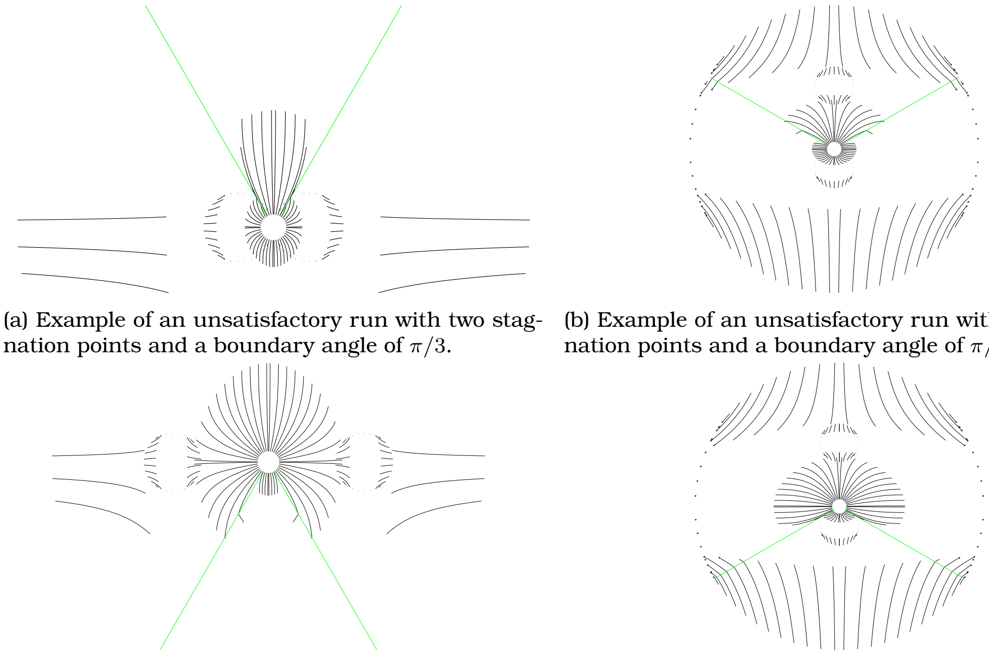
(b) Example of an unsatisfactory run with two stagnation points and a boundary angle of $\pi/6$.

(c) Example of an unsatisfactory run with two stagnation points and a boundary angle of $-\pi/3$.

(d) Example of an unsatisfactory run with two stagnation points and a boundary angle of $-\pi/6$.

Figure 20: Comparison of unsatisfactory runs which include a large number of runs which are not halted by the integration halt. The four landfill examples have a suction level of 5 inWC, a landfill diameter of 25 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$ for the top and bottom sectors respectively.

Figure 21 shows four examples of unsatisfactory runs which are hindered by severe gaps in the integration. Just like Figure 18, there are some apparent trends between vertical and horizontal gaps for Subfigures 21a and 21c, and Subfigures 21b and 21d respectively.



(a) Example of an unsatisfactory run with two stagnation points and a boundary angle of $\pi/3$.

(b) Example of an unsatisfactory run with two stagnation points and a boundary angle of $\pi/6$.

(c) Example of an unsatisfactory run with two stagnation points and a boundary angle of $-\pi/3$.

(d) Example of an unsatisfactory run with two stagnation points and a boundary angle of $-\pi/6$.

Figure 21: Comparison of unsatisfactory runs with major gaps in the pathline integration. The four landfill examples have a suction level of 5 inWC, a landfill diameter of 50 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$ for the top and bottom sectors respectively.

It is important to note that if a computation was given an infinite integration time, it would be possible to produce satisfactory versions of the borderline and unsatisfactory images. However since these cases are examples of completely dysfunctional landfills, the time and effort was not deemed worth the potential result.

4 Arbitrary Number of Sectors

Landfill configurations with three sectors are much more complex than any configuration previously tested. An additional sector adds not only one more contiguity angle, but also one more permeability value. This results in complete freedom of location for the stagnation points, as opposed to being restricted to the vertical and horizontal axes. Moreover, the number of possible stagnation point constellations increases significantly in this chapter. Therefore, the methodology and processing of run results required adaptation by comparison to Chapters 2 and 3.

4.1 Methodology & Code Developement

Transitioning from two to three sectors involves the addition of many parameter options. First, all the values tested in Chapters 2 and 3 for well suction, landfill radius and generation rate were used, however only 4 different permeability combinations were run. These values are listed in Table 4 and due to the complexity of three sectors paired with the issues encountered in Chapters 2 and 3, the lowest permeability tested previously of $1\mathrm{e}-10\,\mathrm{m}^2$ was eliminated. Next, the number of possible permeability arrangements was calculated, so that in a given run for each sector a permutation of all permeabilities is tested. Therefore, as shown in Table 5, it is possible to have six different permeability arrangements for these sectors. The counting convention for where each permeability value is assigned in reference to each boundary is explained in §1.1 and shown in Figure 2

Value of K_1 (in m^2)	**Value of K_2 (in m^2)**	**Value of K_3 (in m^2)**
$1\mathrm{e}-04$	$1\mathrm{e}-05$	$1\mathrm{e}-06$
$1\mathrm{e}-05$	$1\mathrm{e}-06$	$1\mathrm{e}-07$
$1\mathrm{e}-06$	$1\mathrm{e}-07$	$1\mathrm{e}-08$
$1\mathrm{e}-07$	$1\mathrm{e}-08$	$1\mathrm{e}-09$

Table 4: All permeability combinations run for the three sectors case.

Sector 1	Sector 2	Sector 3
K_1	K_2	K_3
K_1	K_3	K_2
K_2	K_1	K_3
K_2	K_3	K_1
K_3	K_2	K_2
K_3	K_2	K_1

Table 5: Possible permeability arrangements for the three sectors case.

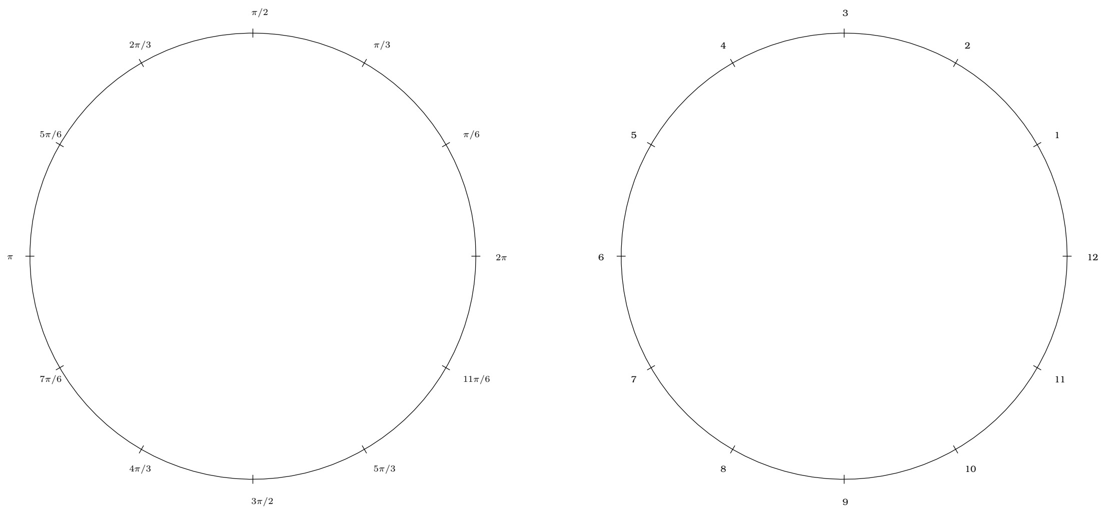


Figure 22: Diagram showing the angles tested (left) and corresponding number designation (right).

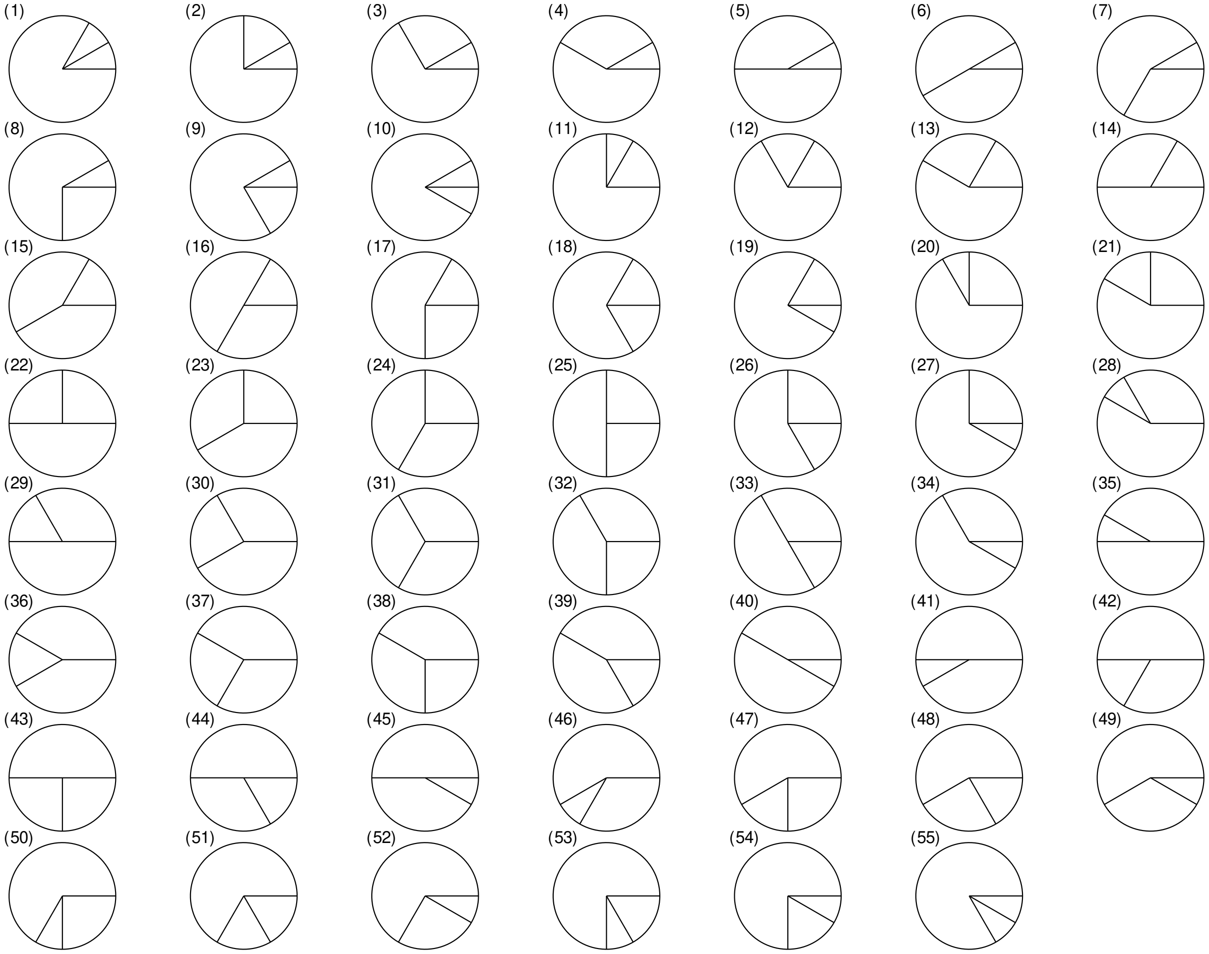


Figure 23: All possible geometries for a landfill with three sectors given the angles available (Figure 22).

In order to run the tests efficiently, it is necessary to decide on the number of different possible boundary angles to be tested. This process started with dividing the landfill into sector units. To create a reasonable coverage of the angle parameter space, a unit of $\pi/6$ was chosen. Figure 22 shows that the angle $\theta = \pi/6$ is at placement 1, and the remaining angles were uniformly spaced and labelled 1 through 12, with 12 corresponding to θ_3, the last contiguity ray. Because of rotational invariance it is possible to keep θ_3 fixed at that value. Figure 23 depicts all distinct individual layouts. Due to the rotational invariance of the governing equations each of these layouts might be arbitrarily rotated as needed in practice. Therefore it was possible to reduce the number of runs by observing that some of these layouts coincide. Table 6 summarizes the result of this process, where the first column shows the number of angle units in each of the sectors, the second column shows which angles were used for the runs in §4.2 and the last column gives the respective layout number from Figure 23.

Sector Size [K_1 K_2 K_3]	Boundary Angle Positions [θ_1 θ_2 θ_3]	Corresponding Number from Figure 23
[1 1 10]	[1 2 12]	1, 10, 55
[1 2 9]	[1 3 12]	2, 9, 11, 19, 53, 54
[1 3 8]	[1 4 12]	3, 8, 20, 27, 50, 52
[1 4 7]	[1 5 12]	4, 7, 28, 34, 46, 49
[1 5 6]	[1 6 12]	5, 6, 35, 40, 41, 45
[2 2 8]	[2 4 12]	12, 18, 21, 26, 51
[2 3 7]	[2 5 12]	13, 17, 47, 48
[2 4 6]	[2 6 12]	14, 16, 29, 33, 42, 44
[2 5 5]	[2 7 12]	15, 36, 39
[3 3 6]	[3 6 12]	22, 25, 43
[3 4 5]	[3 7 12]	23, 24, 30, 32, 37, 38
[4 4 4]	[4 8 12]	31

Table 6: Distinct geometries obtained when taking rotational variance into account.

With all the simplifications complete, the total number of required runs was narrowed down to a manageable number (Table 7).

Landfill Parameters	Original Number of Options	Reduced Number of Options
Number of Angle Geometries	55	12
Permeability Combinations	5	4
Permeability Arrangements	6	6
Configurations Values (p_{out}, r_x, and C)	27	27
Total Runs	44,550	7,776

Table 7: Difference between number of runs before and after parameter reduction.

4.1.1 Calculations & Solutions

The possible stagnation point locations were determined by taking the derivative of Solutions (1.7), with respect to r and θ. Just as in Chapters 2 and 3, these derivatives were set to zero to solve for any existing critical points:

$$\frac{Cr}{k_s}(a^{(s)}\sin(2\theta)+b^{(s)}\cos(2\theta)-1)+\frac{b_2}{r}=0, \tag{4.1a}$$

$$\theta=\frac{1}{2}\arctan\left(\frac{a^{(s)}}{b^{(s)}}\right)+\frac{\pi n}{2}, \qquad n\in\mathbb{Z}. \tag{4.1b}$$

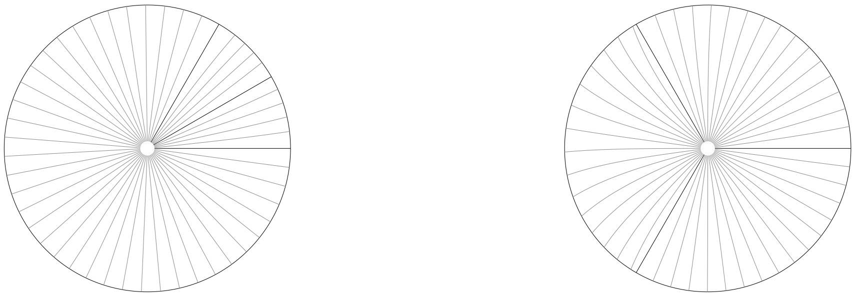
These generate a sequence and Appendix C.9 is used to check how many points out of the sequence fall within a sector.

4.2 Results & Discussion

4.2.1 Classification of Run Success

The criteria used to classify runs in Chaper 4 are different from Chapters 2 and 3. Since a much more complicated system is being handled, the uniform density of pathline distribution was less of a priority and the focus was shifted to whether a run demonstrated the necessary information to understand the flow field induced by a given landfill configuration. The runs were classified into three categories: satisfactory, borderline and unsatisfactory. Any visual without major gaps in integration or visible intersecting lines was deemed satisfactory (Y). From the tests run there were 5,303 satisfactory configurations. A run was classified as borderline (B) when either large gaps were present or pathlines intersected due to insufficient integration accuracy. For three sector configurations there were 953 borderline configurations. Unsatisfactory runs (N) were runs which failed to integrate entirely. There were 1520 such runs. In the companion spreadsheet classifying all runs (`ClassificationOfRunSuccessData`) the following colour code was used to indicate the number of stagnation points in each configuration: none (green), one (yellow), two (orange), three (pink), and five (purple) (Boham, 2023). Constellations of three and five stagnation points were rare and only appeared in low permeability media, and where one of the three sectors had an area much larger than the other two, individually or together.

Runs that were classified as satisfactory essentially followed the same criteria as in previous chapters. These were runs with moderately spaced pathlines, which provide adequate information regarding how the gas is behaving. However, runs with runaway or intersecting lines could not be classified as satisfactory as this suggests an insufficient integration accuracy. Figure 24 shows two examples of satisfactory runs with no stagnation points and therefore examples of two functional landfill collection systems.

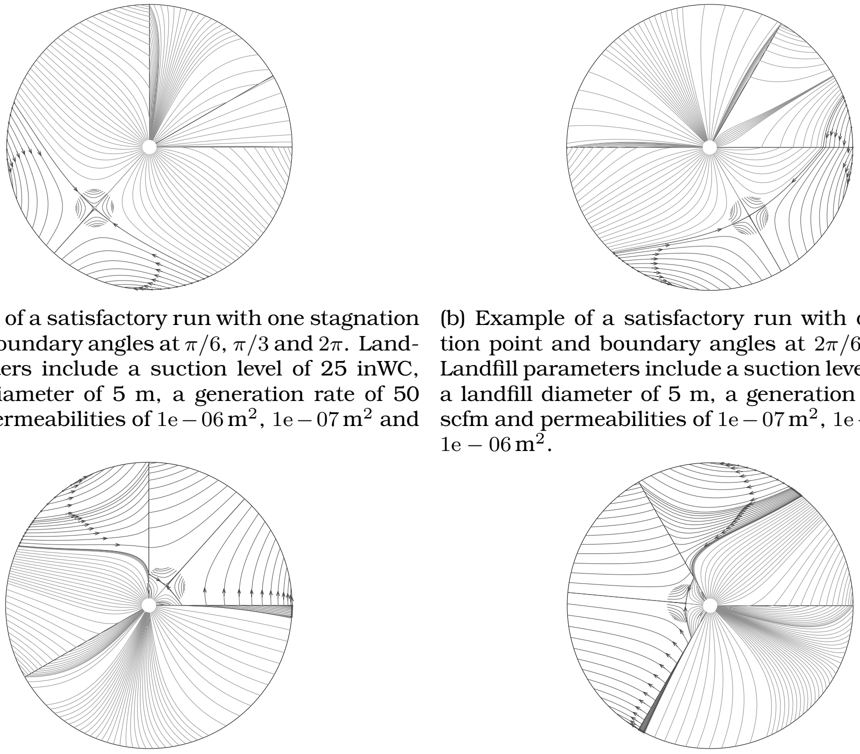


(a) Example of a satisfactory run with zero stagnation points and boundary angles of $\pi/6$, $\pi/3$ and 2π.

(b) Example of a satisfactory run with zero stagnation points and boundary angles of $2\pi/3$, $4\pi/3$ and 2π.

Figure 24: Two examples of satisfactory runs without stagnation points. The two landfill examples have a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-05\,\mathrm{m}^2$, $1\mathrm{e}-06\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$.

Figure 25 shows four examples of satisfactory runs with a single stagnation point. Since the permutation of the permeabilities varies between subfigures, the location of the stagnation point can change dramatically. Subfigure 25a shows an example where the largest sector has the lowest permeability, therefore the one stagnation point present is further away from the well and other pathlines are reasonably uniform. In Subfigure 25b the largest sector has the middle permeability. Here, the stagnation point is in a similar relative position as in Subfigure 25a, however the experience more drastic changes in direction, due to the jump in permeabilities between sectors. Finally Subfigures 25c and 25d show fairly similar examples. Both subfigures have sectors which are close in size and stagnation points very close to the well. The primary difference between the two subfigures is the sectors where the stagnation point is located: sector 2 in Subfigure 25c and sector 3 in 25d, which, in both cases, corresponds to the lowest permeability sector of the three.



(a) Example of a satisfactory run with one stagnation point and boundary angles at $\pi/6$, $\pi/3$ and 2π. Landfill parameters include a suction level of 25 inWC, a landfill diameter of 5 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$.

(b) Example of a satisfactory run with one stagnation point and boundary angles at $2\pi/6$, π and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$, $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-06\,\mathrm{m}^2$.

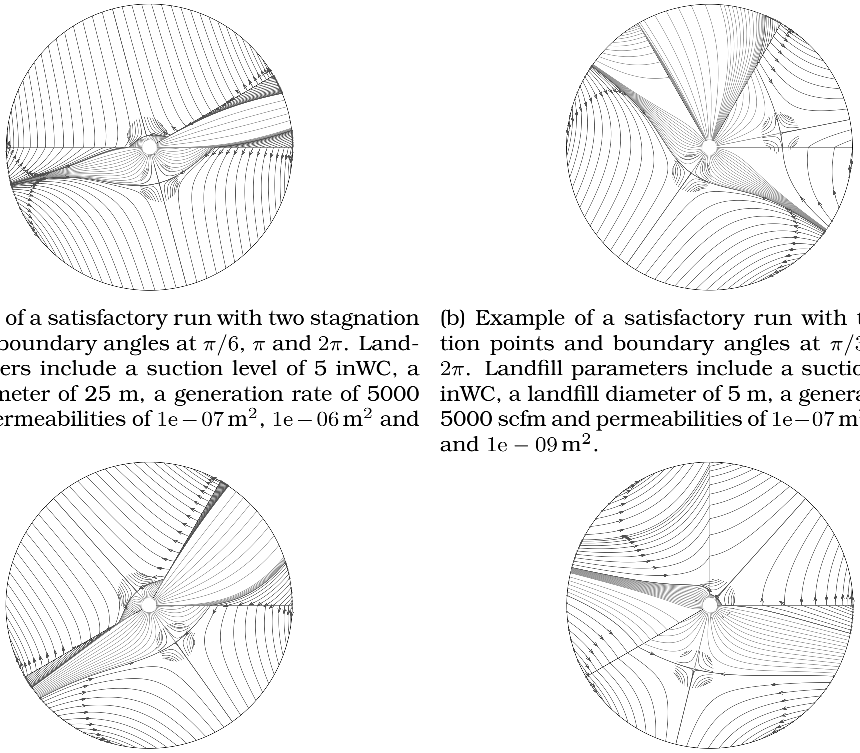
(c) Example of a satisfactory run with one stagnation point and boundary angles at $\pi/2$, $7\pi/6$ and 2π. Landfill parameters include a suction level of 25 inWC, a landfill diameter of 5 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$.

(d) Example of a satisfactory run with one stagnation point and boundary angles at $2\pi/3$, $4\pi/3$ and 2π. Landfill parameters include a suction level of 50 inWC, a landfill diameter of 5 m, a generation rate of 5000 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$.

Figure 25: Examples of satisfactory runs for landfills with three sectors containing one stagnation point.

Figure 26 shows four examples of satisfactory runs with two stagnation points. Similar to what was observed with the one stagnation point fields, due to the nature of the layout geometry and permeability arrangements, a wide range of stagnation point constellations ensues. In

Subfigure 26a the two stagnation points are located in the two sectors with the lower permeabilities. These sectors are also both substantially larger than the third sector, thereby adding additional resistance to the overall system. In contrast, Subfigure 26b shows an example where the stagnation points are located in the two of three sectors with higher permeabilities, whereas the two of three sectors with the lower permeabilities are the smaller sectors. This explains why the stagnation points are not located in the sector with the lowest permeability. Subfigure 26c shows a flow field very similar to that of Subfigure 26a, where the stagnation points are located in the two sectors with lower permeabilities, but in Subfigure 26c the difference of size is less significant. Finally, Subfigure 26d shows an example where the stagnation points are located in the two sectors with the highest and lowest permeabilities, and just as in Subfigure 26c, all sectors are of similar size. The change in resistance between the sectors is thus mainly due to the jump in permeability.



(a) Example of a satisfactory run with two stagnation points and boundary angles at $\pi/6$, π and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 25 m, a generation rate of 5000 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$, $1\mathrm{e}-06\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$.

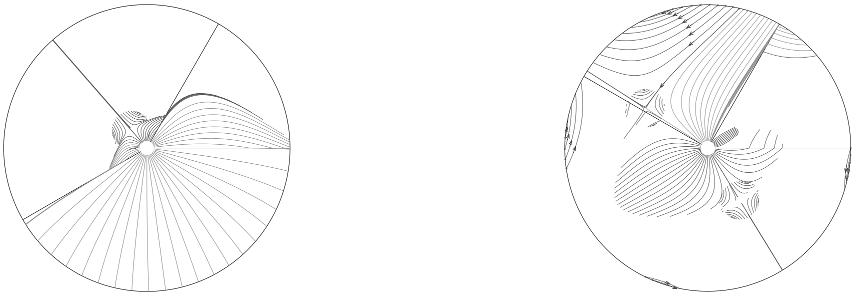
(b) Example of a satisfactory run with two stagnation points and boundary angles at $\pi/3$, $2\pi/3$ and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 5000 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$, $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$.

(c) Example of a satisfactory run with two stagnation points and boundary angles at $\pi/3$, $7\pi/6$ and 2π. Landfill parameters include a suction level of 25 inWC, a landfill diameter of 25 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$.

(d) Example of a satisfactory run with two stagnation points and boundary angles at $\pi/2$, $7\pi/6$ and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 50 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$.

Figure 26: Examples of satisfactory runs for landfills with three sectors containing two stagnation points.

For three sector configurations the classification of borderline runs differed from either of the two sector tests. As mentioned previously, due to the complexity of integrating a three sector flow field, provided that the code was able to generate an image, it was at a minimum deemed borderline. Therefore, runs which had features causing them to previously have been classified as unsatisfactory, such as intersecting pathlines or major irregularities in pathline density, are now being classified as borderline. Figure 27 shows two examples of landfill configurations with significant gaps in coverage. In some examples the gaps are caused by the jump in permeability. In other words, a strong pathline curvature and hence clustering are induced in a small area of the domain, leaving a void in another part, but all lines reach the perimeter. This issue technically could be resolved by increasing the number of pathlines, however this approach would result in pathlines clustering in other regions of the image to an extent that is counter-productive to the purpose of understanding the flow field. In other examples the gaps stem simply from the integration times being too short. Again, this issue could be resolved by increasing the integration time if one was willing to wait for exceptionally long integrations to complete. However, landfill examples that run into this issue are predominantly dysfunctional, and so the wait time is simply not worth the image that would eventually be generated.

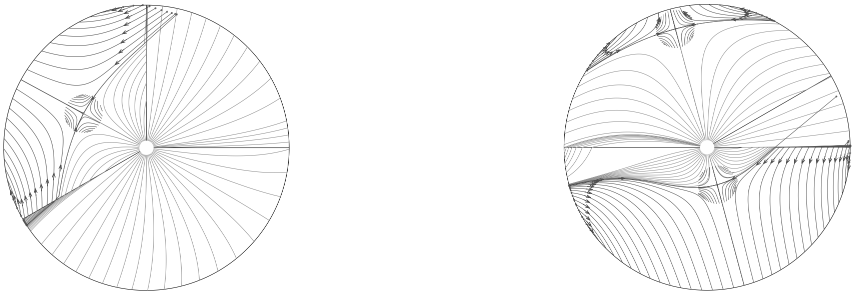


(a) Example of a borderline run with one stagnation point and boundary angles at $\pi/3$, $7\pi/6$ and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 25 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-07\,\mathrm{m}^2$, $1\mathrm{e}-08\,\mathrm{m}^2$ and $1\mathrm{e}-09\,\mathrm{m}^2$.

(b) Example of a borderline run with two stagnation points and boundary angles at $\pi/3$, $5\pi/6$ and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 50 m, a generation rate of 50 scfm and permeabilities of $1\mathrm{e}-08\,\mathrm{m}^2$, $1\mathrm{e}-09\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$.

Figure 27: Example of borderline runs which include moderate gaps within the figure.

Figure 28 shows two examples where the integration accuracy was insufficient, resulting in crossing pathlines. This problem could be resolved by increasing the accuracy of the integration, however, this reintroduces the same cost-benefit calculation between a short wait time and a flawless image.



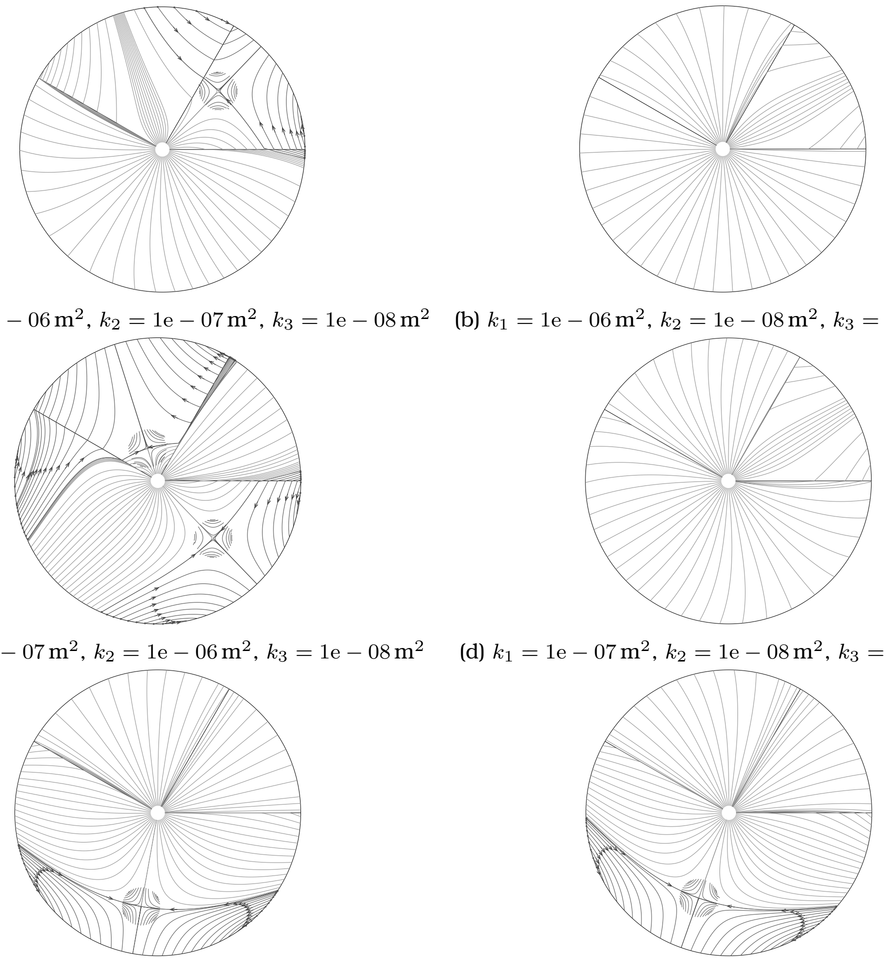
(a) Example of a borderline run with one stagnation point and boundary angles at $\pi/2$, $7\pi/6$ and 2π. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-05\,\mathrm{m}^2$ and $1\mathrm{e}-07\,\mathrm{m}^2$.

(b) Example of a borderline run with two stagnation points and boundary angles at $\pi/6$, π and 2π. Landfill parameters include a suction level of 25 inWC, a landfill diameter of 5 m, a generation rate of 500 scfm and permeabilities of $1\mathrm{e}-09\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$.

Figure 28: Example of borderline runs which exhibit integration failures in the form of overlapping lines.

Any configuration which failed to integrate was classified as unsatisfactory. However, just as with the borderline runs, if one was to wait long enough, it would be possible to integrate these configurations. In such cases the images would likely exhibit one or more of the issues shown in Figures 28 and 27. For this reason, no such figures are included. Nonetheless, it is likely that these runs would result in a flow field representing a landfill functioning so poorly, that they are simply not worth the computational effort.

In summary, a unique characteristic of the three sector configurations occurs when the permeabilities shift between sectors. As depicted in Figures 29 and 30, as the different permeabilities are assigned to sectors of different sizes, the number and location of stagnation points change as well as the nature of the pathlines. In Figure 29 there is a larger discrepancy between the sizes of the largest and smallest sectors. Therefore, when the permeability values shift between sectors, the change in overall resistance acting on the gas is so significant that the number of stagnation points differs between the visuals. In Figure 30 all three sectors are of the same size. Therefore, when the permeability values shift between sectors, the number of stagnation points remains constant. In this case, the feature that is changing is the location of the stagnation point.



(a) $k_1 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-08\,\mathrm{m}^2$

(b) $k_1 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-07\,\mathrm{m}^2$

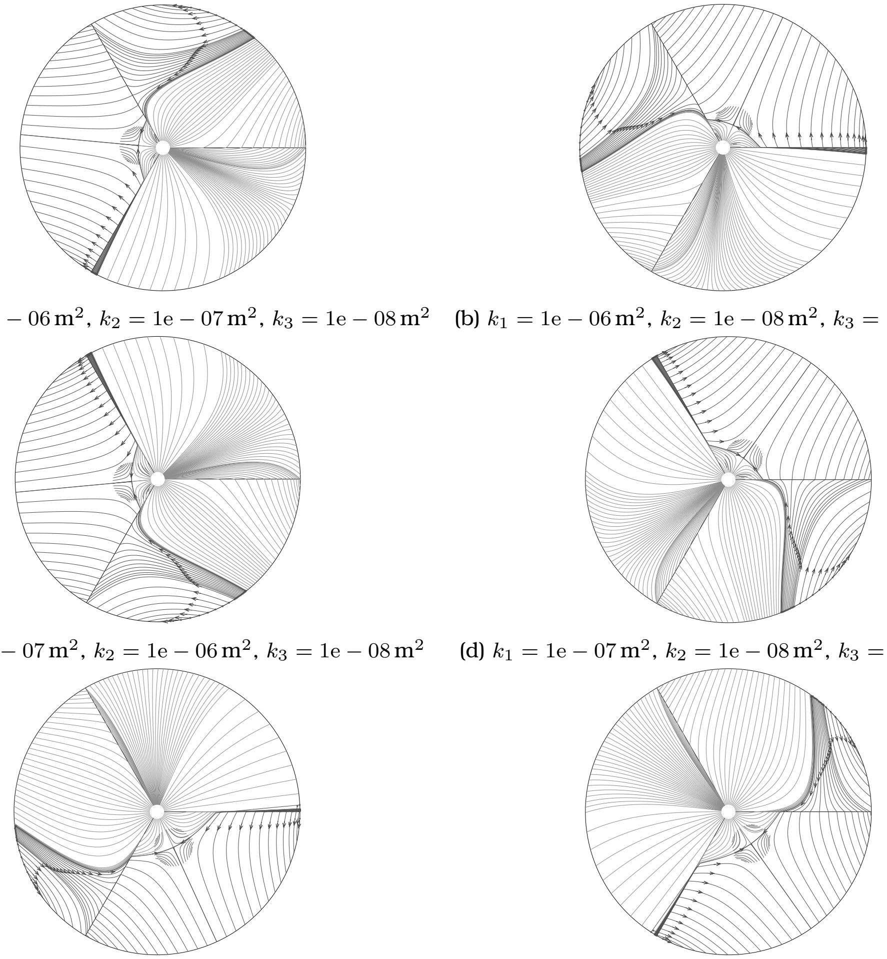
(c) $k_1 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-08\,\mathrm{m}^2$

(d) $k_1 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-06\,\mathrm{m}^2$

(e) $k_1 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-07\,\mathrm{m}^2$

(f) $k_1 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-06\,\mathrm{m}^2$

Figure 29: Effect of permeability rotation on number and location of stagnation points. Landfill parameters include a suction level of 5 inWC, a landfill diameter of 5 m, a generation rate of 50 scfm, permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ and boundary angles of $\pi/3$, $5\pi/6$, and 2π.



(a) $k_1 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-08\,\mathrm{m}^2$ (b) $k_1 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-07\,\mathrm{m}^2$

(c) $k_1 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-08\,\mathrm{m}^2$ (d) $k_1 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-06\,\mathrm{m}^2$

(e) $k_1 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-06\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-07\,\mathrm{m}^2$ (f) $k_1 = 1\mathrm{e}-08\,\mathrm{m}^2$, $k_2 = 1\mathrm{e}-07\,\mathrm{m}^2$, $k_3 = 1\mathrm{e}-06\,\mathrm{m}^2$

Figure 30: Comparison of how rotation of permeabilities affect the number and location of stagnation points. Landfill parameters include a suction level of 50 inWC, a landfill diameter of 5 m, a generation rate of 5000 scfm, permeabilities of $1\mathrm{e}-06\,\mathrm{m}^2$, $1\mathrm{e}-07\,\mathrm{m}^2$ and $1\mathrm{e}-08\,\mathrm{m}^2$ and boundary angles of $2\pi/3$, $4\pi/3$, and 2π.

5 Conclusion

In summary, the objective of this project was to develop a code that could be used by solid waste management industry professionals to understand what landfill parameters allow for an effective collection system. This goal is not to be confused with a comprehensive attempt to map all conceivable – effective or otherwise – landfill configurations.

In Chapters 2 – 4 together, 8,343 different landfill configurations were tested. These spanned a wide range of operational values, resulting in a variety of flow field features and generating images that were classified as satisfactory, borderline, or unsatisfactory. These classifications were an effective way to qualitatively evaluate the efficiency of the code. In this situation a run labelled as unsatisfactory should not be perceived as an undesirable result. In reality, all unsatisfactory and borderline runs correspond to landfill designs that are dysfunctional enough that the practitioners designing a landfill would abandon that design, making production of a satisfactory image unnecessary. The runs classified as satisfactory constitute the group that would be useful to professionals designing landfills, as they encompass the majority of reasonable landfill configurations.

Beyond the physicality of the input, the success of a flow field integration and thence the quality of the produced visual depends on judicious choices of initial points, integration accuracy, and integration length. The tests run in this project were oriented towards understanding and optimizing these choices.

The qualitative analysis determining run success was able to support the understanding of landfill functionality. First, as the well suction increases, the gas flow towards the well becomes more direct in the sense that the pathlines are less curved. By contrast, the lower the well suction, the more likely the presence of stagnation points, which indicates gas escape from the domain. Such a landfill design is therefore not effective. Second, as the landfill radius increases, the higher the chance of stagnation points being present. This can be partially counteracted with an increase in suction, however, based on the values that were observed, this approach will only work to a certain extent and is dependent on the other variables. The presence of a stagnation point sufficiently far from the well is inevitable, although these do not always fall within the physical domain of interest. Third, as the generation rate increases, an increase in well suction is required for the landfill to be effective. Finally, as the permeability of one or more sector decreases, the fluid's ability to move through the solid matrix is also decreased, again resulting in a less effective landfill.

The thousands of visuals examined in this project indicate that the behaviour of the landfill gas flow field is not easy to predict using basic intuition and experience with fluid flow. This problem explains the frequent difficulties encountered by industry professionals when predicting the system response to any engineering changes. In particular, their intuition is often based on the behaviour of liquid flow, such as that in hydraulic wells, but the flow of gas appears to be completely different. The most important observation is that the zone of collection is anything but circular, despite the fact that the well's purpose is to create radial motion and accompanying circular geometry. When working with two sectors, such as those demonstrated in Figure 17, the zone of collection exhibits tapering cusps that are not correlated with the well geometry and therefore cannot be predicted. The shape of the zone of influence varies even more dramatically with three sectors. Another three sector feature, which may be perceived as counter-intuitive, is the unpredictability of the locations of future stagnation points. Despite the availability of closed form solutions, the location of a given stagnation point is almost impossible to infer simply based on the layout and sector permeabilities; extensive calculations are necessary. Finally, another key feature observed was the interplay of sector size and permeability, and the fact that the stagnation points will often, but not always, appear in the sectors with the lower permeabilities.

The code is capable of handling the majority of cases to be encountered in practice, namely two sectors of any angles as well as three sectors. It is expected that engineers and operators represent their system in this qualitative way. The high uncertainties in the parameter values of the waste composition, and the time-dependent level of degradation, render any finer division into smaller subdomains meaningless. The code is implemented in an open source, freely available software. This accessibility is crucial in an industry that is significantly underfunded and not in a position to maintain advanced simulation software or hire professionals trained in its operation.

References

Bear J (1972) Dynamics of Fluids in Porous Media. Dover

Boham M (2023) Classification of Run Success Data. URL https://tru.arcabc.ca/islandora/object/tru%3A6181

Charbeneau R, Street R (1979) Modelling groundwater flow fields containing point singularities: streamlines, travel times and breakthrough curves. Water Resources Research 15:1445 – 1450

Eaton J, Bateman D, Hauberg S, Wehbring R (2018) Free your numbers. GNU Octave, https://www.gnu.org/software/octave/index

Kuhne A, Schack-Kirchner H, Hildebrand E (2012) Gas diffusivity in soils compared to ideal isotropic porous media. Journal of Plant Nutrition and Soil Science 175:34 – 45

Li Y, Cleall P, Ma X, Zhan T, Chen Y (2012) Gas pressure model for layered municipal solid waste landfills. American Society of Civil Engineers 138:752 – 760

Nec Y (2021) Singularities in weakly compressible flow through a porous medium. Fluid Dynamics Research 53:045507

Nec Y, Huculak G (2020) Exact solutions to steady radial flow in a porous medium with variable permeability. Physics of Fluids 32:077108

Negara A, Salama A, Sun S, Elgassier M, Wu YS (2015) Numerical simulation of natural gas flow in anisotropic shale reservoirs. DOI 10.2118/177481-MS

Sahu CK, Flynn MR (2015) Filling box flows in porous media. J Fluid Mech 782:455–478

Speetjens MFM, Wispelaere HNL, van Steenhoven AA (2011) Multi-functional Lagrangian flow structures in three-dimensional AC electro-osmotic micro-flows. Fluid Dynamics Research 43:035503

Wang Q, Gu X, Tang S, Mohammad A, Singh DN, Xie H, Chen Y, Zuo X, Sun Z (2022) Gas transport in landfill cover system: A critical appraisal. Journal of Environmental Management 321

Wikipedia (2023) Permeability (earth sciences). URL https://en.wikipedia.org/wiki/Permeability_(Earth_sciences)

Young A (1989) Mathematical modeling of landfill gas extraction. Journal of Environmental Engineering 115(6):1073–1087

A Code for Semicircular Sectors (Type I)

A.1 Input Processing

Viscosity of a fluid is temperature dependent. For a mixture of gases it is also dependent of the composition. This function calculates viscosity based on temperature and molar fractions and was retained from a previous project (Nec and Huculak, 2020).

```
function mu_mix=visco(T,x)

% base values for Sutherland’s formula

flag=’s’;

if flag==’s’ % Sutherland’s formula

To_CH4=273.15; s_CH4=197.8; muo_CH4=12.01*10^(-6);

To_CO2=293.15; s_CO2=240; muo_CO2=14.8*10^(-6);
%To_CO2=273.15; s_CO2=222.2; muo_CO2=13.7*10^(-6);

To_O2=292.25; s_O2=127; muo_O2=20.18*10^(-6);

To_N2=300.55; s_N2=111; muo_N2=17.81*10^(-6);

mu_CH4=muo_CH4*(T/To_CH4)^1.5*(To_CH4+s_CH4)/(T+s_CH4); % Pa s
mu_CO2=muo_CO2*(T/To_CO2)^1.5*(To_CO2+s_CO2)/(T+s_CO2);
mu_O2=muo_O2*(T/To_O2)^1.5*(To_O2+s_O2)/(T+s_O2);
mu_N2=muo_N2*(T/To_N2)^1.5*(To_N2+s_N2)/(T+s_N2);

elseif flag==’e’ % exponential formula
```

```
% make sure To=293.15K not altered in the main mfile
muo_CH4=13.4*10^(-6); e_CH4=0.87;
muo_CO2=14.8*10^(-6); e_CO2=0.79;
muo_O2=20.0*10^(-6); e_O2=0.69;
muo_N2=17.6*10^(-6); e_N2=0.67;

mu_CH4=muo_CH4*(T/To)^e_CH4;
mu_CO2=muo_CO2*(T/To)^e_CO2;
mu_O2=muo_O2*(T/To)^e_O2;
mu_N2=muo_N2*(T/To)^e_N2;

end


a=3/8; % empirical constant from Thomas A. Davidson's paper - 3/8 or 1/3
       % a=3/8 appears to work negligibly better

%a=0; % works as a flag to have a molar fraction weighted computation

const_read;
mu=[mu_CH4 mu_CO2 mu_O2 mu_N2];

fl=0;
Ymix=sum(sqrt(Mw).*x,2);

if a>0

  for i=1:length(Mw)
  for j=1:length(Mw)
    fl=fl+x(i)*x(j)*Mw(i)^((a+1)/2)*Mw(j)^((a+1)/2)/...
      (sqrt(mu(i)*mu(j))*(Mw(i)+Mw(j))^a);
   end
   end
   fl=2^a*fl/Ymix^2;
   mu_mix=1/fl;

elseif a==0

  mu_mix=sum(Mw.*mu.*x,2)./sum(Mw.*x,2);

end
```

A.2 Universal Constants

This code file defines necessary constants used throughout.

```
K=273.15;
g=9.8067;
ps=101325; ps2=ps^2;  % atmospheric pressure in pascals
Ts=K+15; To=K+20; % standard temperatures
L=0.0065;
Ro=8.3145; % universal, gas constant = Ro/M

i2m_q=0.3048^3/60; %imperial to metric conversion factors
i2m_p=ps/406.8;
i2m_q=0.3048^3/60;
```

```
i2m_l=0.3048;
i2m_d=0.0254;
i2m_p_atm=1000/0.2953;
kg2lb=1/0.4536;

M_CH4=0.016044; %molar weights
M_CO2=0.04401;
M_O2=0.0319988;
M_N2=0.0280134;
M_H2O=0.01801528;
Mw=[M_CH4 M_CO2 M_O2 M_N2];

x_air=[0 0 0.22 0.78]; %molar composition of air
mw_air=sum(x_air.*Mw);
R_air=Ro/mw_air;
```

A.3 Input Parameters

Geometric and physical input parameters; retained from a previous project (Nec and Huculak, 2020).

```
% boundary conditions
bc_flag=1;

elev=0;
p_bar=ps;
p_atm=p_bar*(1-L*elev/Ts)^(g/(R_air*L));
pB=p_atm-0*i2m_p; %px
```

```
p_out=p_atm-p_out*i2m_p; %pp

d=0.5*0.3048;
dh=3/8*0.0254;
nh=2; n=28;
h=[d/2 1 h3 3]';
h=cumsum(h);

% permeability: sectors counting from below the first ray for pN
%k=[1e-07 1e-06];
%k=[1e-08 1e-06]; %figure 4 permeability
%k=[1e-07 1e-06 5e-08 1e-08];  %figure 7
% generation rate m^3/hr, same counting as for k
%C_init=500;

T=15;
x_lfg=[0.5 0.4 0.01];
x_lfg=[x_lfg 1-sum(x_lfg,2)];
R=Ro./sum(x_lfg.*Mw,2);
dl=nh*(dh/2)^2/d;
T=T+K;
mu=visco(T,x_lfg);
C=C_init*ps/(3600*R*To*pi*(h(3)^2-h(1)^2)*dl*n); % no gravel lamina
muRTC=mu*R*T*C;

pX=p_atm;
p_outr=p_out/pX;
pBr=pB/pX;
pX2=pX^2;
```

```
bc_ix=2; % sector # to impose bc
if bc_flag
  a20=pBr^2-p_outr^2+muRTC/(2*k(bc_ix)*pX2)*(h(3)^2-h(1)^2);
  a20=a20/log(h(3)/h(1));
else
  a20=muRTC*h(3)^2/(k(bc_ix)*pX2);
end
a10=p_outr^2+muRTC*h(1)^2/(2*k(bc_ix)*pX2)-a20*log(h(1));
```

A.4 Integration Halts of Semicircular Sectors

This function defines events where the integration is halted for two equal sectors.

```
function [val,term,dir]=p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp)

r=rth(1);
th=rth(2);
x=r*cos(th);
y=r*sin(th);
r_rel=sqrt((x-x_sp)^2+(y-y_sp)^2);

val=r_rel-r_cutoff;
```

A.5 Velocity Field of Semicircular Sectors

This function defines the flow field for the configuration of two equal sectors and top sector radial.

```
function rthp=p2slf(t,rth,th0,int_sign,prm)

const_read

p_out=prm(1);
h3=prm(2);
C_init=prm(3);
k=prm(4:5);

input_read

r=rth(1); th=rth(2);
cth=cos(th); sth=sin(th);
c2th=cos(2*th); s2th=sin(2*th);

K=k/(mu*R*T);
if th0==0
  if th>0
    p=a10+a20*log(r)-r^2*C/(2*K(2)*pX2);
    u=a20/r-r*C/(K(2)*pX2);
    v=0;
  else
    p=a10+a20*log(r)+r^2/(2*pX2)*...
     ((c2th-1)*C/K(1)-c2th*C/K(2));
    u=a20/r+r/pX2*((c2th-1)*C/K(1)-c2th*C/K(2));
    v=-r^2/pX2*(C/K(1)-C/K(2))*s2th;
  end
else
  if th>0
```

```
    b=(C/K(2)-C/K(1))/(1-K(2)/K(1));
    p=a10+a20*log(r)+r^2/(2*pX2)*(b*c2th/cos(2*th0)-C/K(2));
    u=a20/r+r/pX2*(b*c2th/cos(2*th0)-C/K(2));
  else
    b=(C/K(2)-C/K(1))/(K(1)/K(2)-1);
    p=a10+a20*log(r)+r^2/(2*pX2)*(b*c2th/cos(2*th0)-C/K(1));
    u=a20/r+r/pX2*(b*c2th/cos(2*th0)-C/K(1));
  end
  v=-r^2/pX2*b*s2th/cos(2*th0);
end
p=sqrt(p)*pX;
if th>0
  u=-k(2)/mu*u/(2*p)*pX2;
  v=-k(2)/mu*v/(2*p*r)*pX2;
else
  u=-k(1)/mu*u/(2*p)*pX2;
  v=-k(1)/mu*v/(2*p*r)*pX2;
end
rthp=int_sign*[u v/r]'; % r is conversion to polar
```

A.6 Pathline Seeding of Semicircular Sectors

This file defines the starting points of the pathlines being integrated for a configuration with two equal sectors.

```
const_read
input_read
```

```
th0=0;
nth=100;



%well top
r_well = h(3)/20
th_well = 0:pi/20:pi/2;
xunit_well = r_well * cos(th_well);
yunit_well = r_well * sin(th_well);

th_inwell=atan2(yunit_well, xunit_well);
r_inwell=sqrt(xunit_well.^2 + yunit_well.^2);

[x_well,y_well]=fun_int(1500,r_inwell,th_inwell,-1,th0,h(3),1,1,0,0,prm);    % top
%[x_well,y_well]=fun_int(150,r_inwell,th_inwell,-1,th0,h(3),0,1,0,0,prm);

xy_wellt=[x_well y_well];
if flag
  save([’run_wellt_’ file_str ’.txt’],’xy_wellt’,’-ascii’);
else
  plot(x_well, y_well, ’k’, -x_well, y_well, ’k’);
  hold on
  h_dott=plot(x_well(end,:), y_well(end,:), ’.k’, -x_well(end,:), y_well(end,:), ’.k’
end

%well bottom
r_well = h(3)/20
th_well = 3*pi/2:pi/20:2*pi;
```

```
th_well=th_well(1:end-1);
xunit_well = r_well * cos(th_well);
yunit_well = r_well * sin(th_well);

th_inwell=atan2(yunit_well, xunit_well);
r_inwell=sqrt(xunit_well.^2 + yunit_well.^2);

[x_well,y_well]=fun_int(1500,r_inwell,th_inwell,-1,th0,h(3),1,1,0,0,prm);   % bottom
%[x_well,y_well]=fun_int(150,r_inwell,th_inwell,-1,th0,h(3),0,1,0,0,prm);

th_lfend=atan2(y_well(end,2), x_well(end,2))+2*pi;


xy_wellb=[x_well y_well];

if flag
  save(['run_wellb_' file_str '.txt'],'xy_wellb','-ascii');
else
  plot(x_well, y_well, 'k', -x_well, y_well, 'k');
  hold on
  h_dotb=plot(x_well(end,:), y_well(end,:), '.k', -x_well(end,:), y_well(end,:), '.k'
end



th_sp=[0 3*pi/2 ];

%r2_sp=-a20/(muRTC*((1/k(2))-(2/k(1))));   %two equal sectors FIX
r2_sp=-a20./(muRTC*((1/k(1)-1/k(2))*cos(2*th_sp)-1/k(1)))
```

```
%r2_sp=(k(1)/(2*muRTC))*a20;   %two unequal sectors

ix_sp=find(r2_sp>h(1)^2/pX2 & r2_sp<(h(3)*1.001)^2/pX2)
r2_sp=r2_sp(ix_sp);
th_sp=th_sp(ix_sp);

r_sp=sqrt(r2_sp)*pX;

x_sp=r_sp.*cos(th_sp);
y_sp=r_sp.*sin(th_sp);



%stagnation point top
X_stpt=[];
Y_stpt=[];
X_stpb=[];
Y_stpb=[];

for i=1:length(ix_sp)



if abs(x_sp(i))<10^(-12)
  th_circle = 0:pi/15:pi/2;
else
  th_circle = 0:pi/15:pi;
end

r_circle = h(3)/7.5
xunit = r_circle * cos(th_circle) + x_sp(i);
```

```
yunit = r_circle * sin(th_circle) + y_sp(i);

th_in=atan2(yunit, xunit);
r_in=sqrt(xunit.^2 + yunit.^2);

ix_in=find(r_in>r_well);
th_in=th_in(ix_in);
r_in=r_in(ix_in);


[x_stpt,y_stpt]=fun_int(1000,r_in,th_in,-1,th0,r_circle*1.01,1,1,x_sp(i),y_sp(i),prm)
%[x_stpt,y_stpt]=fun_int(40,r_in,th_in,-1,th0,r_circle*1.01,0,1,x_sp(i),y_sp(i),prm);

X_stpt=[X_stpt x_stpt];
Y_stpt=[Y_stpt y_stpt];

%stagnation point bottom

if abs(x_sp(i))<10^(-12)
  th_circle = 3*pi/2:pi/15:2*pi;
else
  th_circle = pi:pi/15:2*pi;
end

%th_circle = 3*pi/2:pi/15:2*pi;
th_circle=th_circle(2:end);
r_circle = h(3)/7.5
xunit = r_circle * cos(th_circle) + x_sp(i);
yunit = r_circle * sin(th_circle) + y_sp(i);
```

```
th_in=atan2(yunit, xunit);
r_in=sqrt(xunit.^2 + yunit.^2);

ix_in=find(r_in>r_well);
th_in=th_in(ix_in);
r_in=r_in(ix_in);

[x_stpb,y_stpb]=fun_int(1000,r_in,th_in,-1,th0,r_circle*1.01,1,1,x_sp(i),y_sp(i),prm)
%[x_stpb,y_stpb]=fun_int(40,r_in,th_in,-1,th0,r_circle*1.01,0,1,x_sp(i),y_sp(i),prm);

X_stpb=[X_stpb x_stpb];
Y_stpb=[Y_stpb y_stpb];



end

if ~isempty(ix_sp)
xy_stpt=[X_stpt Y_stpt];
if flag
  save([’run_stpt_’ file_str ’.txt’],’xy_stpt’,’-ascii’);
else
  plot(X_stpt, Y_stpt, ’k’, -X_stpt, Y_stpt, ’k’);
  hold on
  %h_dotpt=plot(X_stpt(end,:), Y_stpt(end,:), ’.k’, -X_stpt(end,:), Y_stpt(end,:), ’.
end
xy_stpb=[X_stpb Y_stpb];
if flag
  save([’run_stpb_’ file_str ’.txt’],’xy_stpb’,’-ascii’);
```

```
else
  plot(X_stpb, Y_stpb, 'k', -X_stpb, Y_stpb, 'k');
  hold on
  %h_dotpb=plot(X_stpt(end,:), Y_stpt(end,:), '.k', -X_stpt(end,:), Y_stpt(end,:), '.
end



%landfill
th_lf = 3*pi/2:pi/30:th_lfend;
th_lf=th_lf(1:end-1);
r_lf = h(3)
xunit_lf = r_lf * cos(th_lf);
yunit_lf = r_lf * sin(th_lf);

th_inlf=atan2(yunit_lf, xunit_lf);
r_inlf=sqrt(xunit_lf.^2 + yunit_lf.^2);



[x_lf,y_lf]=fun_int(3000,r_inlf,th_inlf,-1,th0,h(3)*1.001,1,1,0,0,prm);
%[x_lf,y_lf]=fun_int(30,r_inlf,th_inlf,-1,th0,h(3)*1.001,0,1,0,0,prm);

xy_lf=[x_lf y_lf];

if flag
  save(['run_lf_' file_str '.txt'],'xy_lf','-ascii');
else
  plot(x_lf, y_lf, 'k', -x_lf, y_lf, 'k');
  hold on
  h_dotlf=plot(x_lf(end,:), y_lf(end,:), '.k', -x_lf(end,:), y_lf(end,:), '.k');  %cr
```

```
end


end


plot( cos(th0)*[r_well h(3)], sin(th0)*[r_well h(3)], 'g' )
plot( cos(pi)*[r_well h(3)], sin(pi)*[r_well h(3)], 'g' )

axis equal
```

A.7 Integration: Preliminaries for Semicircular Sectors

This function collates integrated pathlines.

```
function [x,y]=fun_int(t_span,r_in,th_in,int_sign,th0,r_cutoff,term,dir,x_sp,y_sp,prm

x=[];
y=[];
n=50;

for i=1:length(r_in)
  rth_in=[r_in(i) th_in(i)];
  [t,rth]=newSL([],rth_in,th0,r_cutoff,t_span, int_sign,term,dir,x_sp,y_sp,prm);
  t1=linspace(t(1),t(end),n)';
  x1=rth(:,1).*cos(rth(:,2));
  y1=rth(:,1).*sin(rth(:,2));
  x=[x interp1(t,x1,t1)];
  y=[y interp1(t,y1,t1)];
end
```

```
%last update 05/05/2021
```

A.8 Integration of Semicircular Sectors

This function calls ode45, Octave's main integration facility.

```
function [t_out,rth_out]=newSL(t_in,rth_in,th0,r_cutoff,dlt,int_sign,term,dir,x_sp,y_

if isempty(t_in)
  tspan=[0 dlt];
  rth_init=rth_in;
  rth_in=[];
else
  tspan=[t_in(end) t_in(end)+dlt];
  rth_init=rth_in(end,:);
end

if term==1
  %opt=odeset('events',@(t,rth) p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp));
  opt=odeset('events',@(t,rth) p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp),'AbsTol',1e-0
else
  opt=odeset('events', []);
end

[t_add,rth_add,te,rthe,ie]=ode45(@(t,rth) p2slf(t,rth,th0,int_sign,prm),tspan,rth_ini
t_out=[t_in;t_add];
rth_out=[rth_in;rth_add];
```

A.9 Main Running File of Semicircular Sectors

Main running file.

```
clear all
close all
warning off

flag=0;

Input=load(’input.txt’);
n=size(Input,1);

for i=1:n
try
  i
  tic
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  file_str=[num2str(p_out) ’_’ num2str(h3) ’_’ num2str(C_init) ’_’ num2str(-log10(k(1
  figure(1)
  p2sl
  toc
catch
  display(['config ' num2str(prm) ' skipped'])
  continue
end

end

%p2sl_surf(th0,prm,2)
```

A.10 Plotting Semicircular Sectors

This file plots visuals from saved integration files.

```
clear all
close all
warning off

Input=load('input.txt');
n=size(Input,1);

for i=1:n
try
  i
  tic
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  file_str=[num2str(p_out) '_' num2str(h3) '_' num2str(C_init) '_' num2str(-log10(k(1
  p2sl
  toc
catch
  display(['config ' num2str(prm) ' skipped'])
  continue
end
end

%figure(2) %change number for new window

Input=load('input.txt');
n=size(Input,1);

frac=0.01;

File_str='';

for i=1:n
  figure(i)
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  file_str=['_' num2str(p_out) '_' num2str(h3) '_' num2str(C_init) '_' num2str(-log10
  File_str=strvcat(File_str, file_str);
  set(gcf,’name’,file_str)
  eval([’L=ls(’ file_str ’*’’)’]);
  m=size(L,1);
  for j=1:m
    xy=dlmread(deblank(L(j,:)));
    n_xy=size(xy,2)/2;
    L_sl=(xy(1,1:n_xy)-xy(end,1:n_xy)).^2 + (xy(1,n_xy+1:end)-xy(end,n_xy+1:end)).^2;
    L_sl=sqrt(L_sl);
    L_slmin=min(L_sl);
    L_slmax=max(L_sl);
    if L_slmin/L_slmax<2*frac
      ix_sl=find(L_sl>L_slmin+frac*(L_slmax-L_slmin));
    else
      ix_sl=1:n_xy;
    end
    %plot(xy(:,1:n_xy),xy(:,n_xy+1:end))
    plot(xy(:,ix_sl),xy(:,n_xy+ix_sl),’k’,-xy(:,ix_sl),xy(:,n_xy+ix_sl),’k’)
    hold on
  end
  axis off
end

hold on
```

B Code for Semicircular Sectors (Type II) and Sectors of Two Sectors of Unequal Angles

B.1 Integration Halts of Two Sectors of Unequal Angles

This function defines events where the integration is halted for two unequal sectors.

```
function [val,term,dir]=p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp)

r=rth(1);
th=rth(2);
x=r*cos(th);
y=r*sin(th);
r_rel=sqrt((x-x_sp)^2+(y-y_sp)^2);

val=r_rel-r_cutoff;
term=repmat(term,length(val),1);
dir=repmat(dir,length(val),1);
```

B.2 Velocity Field of Two Sectors of Unequal Angles

This function defines the flow field for the configuration of two unequal sectors.

```
function rthp=p2slf(t,rth,thc,int_sign,prm)

const_read

p_out=prm(1);
```

```
h3=prm(2);
C_init=prm(3);
k=prm(4:5);

input_read

r=rth(1); th=rth(2);
cth=cos(th); sth=sin(th);
c2th=cos(2*th); s2th=sin(2*th);

while th<thc(1)
  th=th+2*pi;
end

while th>thc(end)
  th=th-2*pi;
end

if ~(th>=thc(1) & th<=thc(end))
  rth
  display('cannot delimit th in (th0, th0+2pi)')
  return
end

for i=1:length(thc)
  if abs(th-thc(i))<1e-12
    ix=i;
    break
  end
```

```
end
for i=1:(length(thc)-1)
  if (th>thc(i) & th<thc(i+1))
    ix=i;
    break
  end
end

K=k/(mu*R*T);
if 1==0
%if thc(1)==0
  if ix==1
    p=a10+a20*log(r)-r^2*C/(2*K(2)*pX2);
    u=a20/r-r*C/(K(2)*pX2);
    v=0;
  elseif ix==2
    p=a10+a20*log(r)+r^2/(2*pX2)*...
     ((c2th-1)*C/K(1)-c2th*C/K(2));
    u=a20/r+r/pX2*((c2th-1)*C/K(1)-c2th*C/K(2));
    v=-r^2/pX2*(C/K(1)-C/K(2))*s2th;
  end
end
%else
  if ix==1
    p=a10+a20*log(r)+C*r^2/(2*pX2*K(2))*(c2th/cos(2*thc(1))-1);
    u=a20/r+C*r/(pX2*K(2))*(c2th/cos(2*thc(1))-1);
    v=-C*r^2/(pX2*K(2))*s2th/cos(2*thc(1));
  else
    p=a10+a20*log(r)+C*r^2/(2*pX2*K(1))*(c2th/cos(2*thc(1))-1);
    u=a20/r+C*r/(pX2*K(1))*(c2th/cos(2*thc(1))-1);
    v=-C*r^2/(pX2*K(1))*s2th/cos(2*thc(1));
  end
%end
p=sqrt(p)*pX;
if ix==1
  u=-k(2)/mu*u/(2*p)*pX2;
  v=-k(2)/mu*v/(2*p*r)*pX2;
else
  u=-k(1)/mu*u/(2*p)*pX2;
  v=-k(1)/mu*v/(2*p*r)*pX2;
end
rthp=int_sign*[u v/r]’;
```

B.3 Pathline Seeding of Two Sectors of Unequal Angles

This file defines the starting points of the pathlines being integrated for a configuration with two unequal sectors.

```
const_read
input_read
nth=100;
thc=[th0, pi-th0, th0+2*pi];
parcel_ix=-(1:5);
R_init=cell(5,1);
Th_init=cell(5,1);
x_int=cell (5,1);
y_int=cell (5,1);
```

```
parcel_ix(1)=-parcel_ix(1);

r_well = h(3)/20
%th_well = 0:pi/20:pi/2;   %integration from 0 to pi/2
%th_well = pi/2:pi/20:3*pi/2;    %integration from th0 to pi/2
th_well=[linspace(0.51*pi,0.995*pi,11) linspace(1.005*pi,1.49*pi,11)];
xunit_well = r_well * cos(th_well);
yunit_well = r_well * sin(th_well);

th_inwell=th_well;
r_inwell=r_well*ones(1,length(th_inwell));

R_init{1}=r_inwell;
Th_init{1}=th_inwell;

r_circle = h(3)/7.5

t_span=[1500 1000 3000 3000 3000];
r_cutoff=[h(3) r_circle*1.01 h(3)*1.003 h(3)*1.003 h(3)*1.003];
r_cutoff=[r_cutoff; 1.5*h(3)*ones(1,length(r_cutoff))];
term=[1 1 1 1 1];
dir=[1 1 1 1 1];
x_ref=cell(1,5);
y_ref=cell(1,5);
x_ref([1 3:5])=0;
y_ref([1 3:5])=0;

[x_well,y_well]=fun_int(1500,r_inwell,th_inwell,-1,thc,r_cutoff(:,1),term(1),dir(1),x
x_int{1}=x_well;
y_int{1}=y_well;

th_lfend_t=atan2(y_well(end,11), x_well(end,11));
th_lfend_t=th_lfend_t+2*pi*(th_lfend_t<0);
th_lfend_b=atan2(y_well(end,13), x_well(end,13));
th_lfend_b=th_lfend_b+2*pi*(th_lfend_b<0);


th_sp=[0 pi/2 3*pi/2 ];       %if you swap the order of th_sp change the order of k i
k_sp=[k(1)*(th0>0)+k(2)*(th0<0) k(2) k(1)];        %[k(1) k(2) k(1)] matches the orde
for i=1:length(th_sp)
  if(th0==0 & th_sp(i)==0)
    % horizontal points for two equal sectors with purely radial top
    %r2_sp(i)=-a20./(muRTC*((1/k(1)-1/k(2))*cos(2*th_sp(i))-1/k(1)));
    % horizontal points for two equal sectors symmetric top-bottom
    r2_sp(i)=Inf;
  else
    % everything else (two unequal and two equal vertical)
    r2_sp(i)=a20*k_sp(i)./(muRTC*(1-cos(2*th_sp(i))/cos(2*th0)));
  end
end
ix_sp=find(r2_sp>h(1)^2/pX2 & r2_sp<(h(3)*1.001)^2/pX2)
r2_sp=r2_sp(ix_sp);
th_sp=th_sp(ix_sp);
r_sp=sqrt(r2_sp)*pX;
x_sp=r_sp.*cos(th_sp);
y_sp=r_sp.*sin(th_sp);
```

```
x_ref{2}=x_sp;
y_ref{2}=y_sp;

Th_init{2,1}=pi/30:pi/15:(2-1/30)*pi;
Th_init{2,2}=pi/2:pi/15:3*pi/2;
Th_init{2,3}=pi/2:pi/15:3*pi/2;

l_sp=length(x_sp);
if l_sp>0
  parcel_ix(2)=-parcel_ix(2);
end

for i=1:l_sp
  xunit = r_circle * cos(Th_init{2,ix_sp(i)}) + x_sp(i);
  yunit = r_circle * sin(Th_init{2,ix_sp(i)}) + y_sp(i);
  th_in=atan2(yunit, xunit);
  r_in=sqrt(xunit.^2 + yunit.^2);
  ix_in=find(r_in>r_well);
  th_in=th_in(ix_in);
  r_in=r_in(ix_in);
  R_init{2,ix_sp(i)}=r_in;
  Th_init{2,ix_sp(i)}=th_in;
end

if (length(ix_sp)==1 && ix_sp(1)==3) || (length(ix_sp)==2 && ix_sp(2)==3) || (length(
  parcel_ix(3)=-parcel_ix(3);
end

%th_lf = th_lfend_b:pi/30:3*pi/2;
```

```
th_lf=linspace(th_lfend_b, 1.49*pi, 16);
%th_lf=th_lf(1:end-1);
r_lf = h(3)
xunit_lf = r_lf * cos(th_lf);
yunit_lf = r_lf * sin(th_lf);

th_inlf=atan2(yunit_lf, xunit_lf);
r_inlf=sqrt(xunit_lf.^2 + yunit_lf.^2);

R_init{3}=r_inlf;
Th_init{3}=th_inlf;

if (length(ix_sp)==2 && ix_sp(1)==2 && ix_sp(2)==3) || (length(ix_sp)==3)
  parcel_ix(4)=-parcel_ix(4);
end

th_lf = 0.51*pi:pi/30:th_lfend_t;
%th_lf=th_lf(1:end-1);
r_lf = h(3)
xunit_lf = r_lf * cos(th_lf);
yunit_lf = r_lf * sin(th_lf);

th_inlf=atan2(yunit_lf, xunit_lf);
r_inlf=sqrt(xunit_lf.^2 + yunit_lf.^2);

R_init{4}=r_inlf;
Th_init{4}=th_inlf;

if length(ix_sp)==1 && ix_sp(1)==1
```

```
  parcel_ix(5)=-parcel_ix(5);
end



th_lf = th_lfend_t:pi/30:th_lfend_b;
th_lf=th_lf(1:end-1);
r_lf = h(3)
xunit_lf = r_lf * cos(th_lf);
yunit_lf = r_lf * sin(th_lf);

th_inlf=atan2(yunit_lf, xunit_lf);
r_inlf=sqrt(xunit_lf.^2 + yunit_lf.^2);

R_init{5}=r_inlf;
Th_init{5}=th_inlf;



p_ix=find(parcel_ix>0);
for i=p_ix
for j=1:(1*(i~=2)+l_sp*(i==2))
  if i~=2
     jx=j;
  elseif ~isempty(ix_sp)
     jx=ix_sp(j);
  end
  if i>1
    [x_int{i,j},y_int{i,j}]=fun_int(t_span(i),R_init{i,jx},Th_init{i,jx},-1,thc, r_cu
  end
  xy=[x_int{i,j} y_int{i,j}];
  if flag
    save([’run_parcel’ num2str(i) ’_’ file_str ’.txt’],’xy’,’-ascii’);
  else
    plot(x_int{i,j}, y_int{i,j}, ’k’, -x_int{i,j}, y_int{i,j}, ’k’);
    hold on

    x_dot=x_int{i,j}(end,:);
    y_dot=y_int{i,j}(end,:);
    r_dot=sqrt(x_dot.^2+y_dot.^2);
    ix_dot=find(r_dot>0.9*h(3));
    x_dot=x_dot(ix_dot);
    y_dot=y_dot(ix_dot);

    h_dot=plot(x_dot, y_dot, ’.k’, -x_dot, y_dot, ’.k’);  %creates dot at the end of

  end
end
end

plot( cos(thc(1))*[r_well h(3)], sin(thc(1))*[r_well h(3)], ’g’ )
plot( cos(thc(2))*[r_well h(3)], sin(thc(2))*[r_well h(3)], ’g’ )

axis equal
```

B.4 Integration: Preliminaries for Two Sectors of Unequal Angles

```
function [x,y]=fun_int(t_span,r_in,th_in,int_sign,thc,r_cutoff,term,dir,x_sp,y_sp,prm

x=[];
y=[];
n=50;

for i=1:length(r_in)
  rth_in=[r_in(i) th_in(i)];
  [t,rth]=newSL([],rth_in,thc,r_cutoff,t_span, int_sign,term,dir,x_sp,y_sp,prm);
  t1=linspace(t(1),t(end),n)';
  x1=rth(:,1).*cos(rth(:,2));
  y1=rth(:,1).*sin(rth(:,2));
  x=[x interp1(t,x1,t1)];
  y=[y interp1(t,y1,t1)];
end


%last update 05/05/2021
```

B.5 Integration of Two Sectors of Unequal Angles

This function calls ode45, Octave's main integration facility.

```
function [t_out,rth_out]=newSL(t_in,rth_in,thc,r_cutoff,dlt,int_sign,term,dir,x_sp,y_
if isempty(t_in)
  tspan=[0 dlt];
  rth_init=rth_in;
  rth_in=[];
else
  tspan=[t_in(end) t_in(end)+dlt];
  rth_init=rth_in(end,:);
end

if term==1
  %opt=odeset('events',@(t,rth) p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp));
  opt=odeset('events',@(t,rth) p2sle(t,rth,r_cutoff,term,dir,x_sp,y_sp),'AbsTol',1e-0
else
  opt=odeset('events', []);
end

[t_add,rth_add,te,rthe,ie]=ode45(@(t,rth) p2slf(t,rth,thc,int_sign,prm),tspan,rth_ini

t_out=[t_in;t_add];
rth_out=[rth_in;rth_add];
```

B.6 Main Running File of Two Sectors of Unequal Angles

Main running file.

```
clear all
close all
warning off
```

```
flag=0;

Input=load('input.txt');
n=size(Input,1);

nix=1:n;
%nix=sort([1:3:n 2:3:n]);
%nix=sort([1:3:n]);

for i=nix
%try
  i
  tic
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  if prm(6)==0
    th0=0;
  else
    th0=1/prm(6)*pi;
  end
 if th0>=0
  th_sign='p';
 else
  th_sign='n';
 end
```

```
  file_str=[num2str(p_out) '_' num2str(h3) '_' num2str(C_init) '_' num2str(-log10(k(1
  figure(i)
  p2sl
  toc
%catch
  display(['config ' num2str(prm) ' skipped'])
  continue
%end

end

%p2sl_surf_sim(th0,prm,2)
```

B.7 Plotting Two Sectors of Unequal Angles

This file plots visuals from saved integration files.

```
clear all
close all
warning off

Input=load('input.txt');
n=size(Input,1);

for i=1:n
try
  i
  tic
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  file_str=[num2str(p_out) ’_’ num2str(h3) ’_’ num2str(C_init) ’_’ num2str(-log10(k(1
  p2sl
  toc
catch
  display([’config ’ num2str(prm) ’ skipped’])
  continue
end
end

%figure(2) %change number for new window

Input=load(’input.txt’);
n=size(Input,1);

frac=0.01;

File_str=’’;

for i=1:n
  figure(i)
  prm=Input(i,:);
  p_out=prm(1);
  h3=prm(2);
  C_init=prm(3);
  k=prm(4:5);
  file_str=[’_’ num2str(p_out) ’_’ num2str(h3) ’_’ num2str(C_init) ’_’ num2str(-log10
  File_str=strvcat(File_str, file_str);
  set(gcf,’name’,file_str)
  eval([’L=ls(’ file_str ’*’’)’]);
  m=size(L,1);
  for j=1:m
    xy=dlmread(deblank(L(j,:)));
    n_xy=size(xy,2)/2;
    L_sl=(xy(1,1:n_xy)-xy(end,1:n_xy)).^2 + (xy(1,n_xy+1:end)-xy(end,n_xy+1:end)).^2;
    L_sl=sqrt(L_sl);
    L_slmin=min(L_sl);
    L_slmax=max(L_sl);
    if L_slmin/L_slmax<2*frac
      ix_sl=find(L_sl>L_slmin+frac*(L_slmax-L_slmin));
    else
      ix_sl=1:n_xy;
    end
    %plot(xy(:,1:n_xy),xy(:,n_xy+1:end))
    plot(xy(:,ix_sl),xy(:,n_xy+ix_sl),’k’,-xy(:,ix_sl),xy(:,n_xy+ix_sl),’k’)
    hold on
  end
  axis off
end

hold on
```

C Code for Arbitrary Number of Sectors

C.1 Coefficients in Equation (4.1)

This function defines the linear system that gives the $a^{(s)}$ and $b^{(s)}$ coefficients in Equation (1.7); retained from a previous project (Nec and Huculak, 2020).

```
function Cm=Cm_m(thc,m,kr,N)

Cm=zeros(2*N,2*N);
cmth=cos(m*thc);
smth=sin(m*thc);
for i=1:N-1
  Cm(2*i-1,2*i+(-1:2))=[smth(i) cmth(i) -smth(i)*kr(i) -cmth(i)*kr(i)];
  Cm(2*i,2*i+(-1:2))=[cmth(i) -smth(i) -cmth(i) smth(i)];
end
Cm(end-1,1:2)=-kr(N)*[smth(N) cmth(N)];
Cm(end,1:2)=[-cmth(N) smth(N)];
Cm(end-1,end-1:end)=[smth(N) cmth(N)];
Cm(end,end-1:end)=[cmth(N) -smth(N)];
```

C.2 Integration Halts of Arbitrary Sectors

This function defines events where the integration is halted for three sectors.

```
function [val,trm,dir]=psle(t,rth,r_trm,x0,y0,int_sign)

r=rth(1);
th=rth(2);
```

```
x=r*cos(th);
y=r*sin(th);
r_rel=sqrt((x-x0)^2+(y-y0)^2);

val=[r_rel-r_trm; imag(rth)];
lv=length(val);
trm=ones(lv,1);
dir=int_sign*trm;
dir(end-1:end)=[0 0]';
```

C.3 Velocity Field of Arbitrary Sectors

This function defines the flow field for the configuration with three sectors.

```
function rthp=pslf(t,rth,thc,int_sign,k,muRTC,mu,pX,acs0)

r=rth(1); th=rth(2);
th=th_dlm(th,thc(1),thc(end),2*pi);
if ~(th>=thc(1) & th<=thc(end))
  [thc(1) th thc(end)]/pi
  display('Cannot delimit th in (0, 2\pi)')
end
ix=[];
for i=1:length(thc)
  if abs(th-thc(i))<1e-12
    ix=i;
    break
  end
```

```
end
for i=1:(length(thc)-1)
  if (th>thc(i) && th<thc(i+1))
    ix=i+1;
    break
  end
end
N=length(k);
if ix==N+1
  ix=1;
end

a10=acs0(1); a20=acs0(2);
acs=acs0(3:end);
a=acs(1:2:end); b=acs(2:2:end);
pX2=pX^2;
c2th=cos(2*th); s2th=sin(2*th);
p=a10+a20*log(r)+muRTC*r^2/(2*pX2*k(ix))*(a(ix)*s2th+b(ix)*c2th-1);
u=a20/r+muRTC*r/(pX2*k(ix))*(a(ix)*s2th+b(ix)*c2th-1);
v=muRTC*r^2/(pX2*k(ix))*(a(ix)*c2th-b(ix)*s2th);
p=sqrt(p)*pX;
u=-k(ix)/mu*u/(2*p)*pX2;
v=-k(ix)/mu*v/(2*p*r)*pX2;
rthp=int_sign*[u v/r]’; % u=dr/dt, v=r d\th/dt
```

C.4 Pathline Seeding of Arbitrary Sectors

This file defines the starting points of the pathlines being integrated for a configuration with three sectors.

```
global x_int y_int
x_int=[]; y_int=[]; Th_init=[]; R_init=[];

pi12=th_dlm(pi/2,thc(1),thc(end),2*pi);
pi32=th_dlm(3*pi/2,thc(1),thc(end),2*pi);
pi_tol=pi*1e-12;
sp_tol=pi*1e-4;

[th_sp,r_sp]=sploc(thc,k,muRTC,acs0);
ix_sp=find(r_sp>d/(2*pX) & r_sp<h*1.001/pX);
r_sp=r_sp(ix_sp);
th_sp=th_sp(ix_sp);
r_sp=r_sp*pX;
x_sp=r_sp.*cos(th_sp);
y_sp=r_sp.*sin(th_sp);
if N==2
  th_sp(find(abs(th_sp-pi12)<pi_tol))=pi12;
  th_sp(find(abs(th_sp-pi32)<pi_tol))=pi32;
  x_sp=0*x_sp;
end
display('(x,y) coordinates of s.p.')
[x_sp y_sp]
l_sp=length(x_sp);

r_trm=[h r_circle h r_well;
       1.1*h 1.1*r_circle 1.1*h 0.9*r_well];
num_min=3;
dthf=diff(thc)/(2*pi);
```

```
thc_num=round(dthf*num);
ix_num=find(thc_num<num_min);
thc_num(ix_num)=num_min;
th_well=[];
dthc=diff(thc);
for i=1:N
  ix_sp=find(th_sp>=thc(i) & th_sp<thc(i+1));
  if isempty(ix_sp)
    th_tmp=linspace(thc(i),thc(i+1),thc_num(i)+2);
  else
    th1=thc(i);
    th2=th_sp(ix_sp(1))-sp_tol;
    dth12=th2-th1;
    th_tmp=linspace(th1,th2,round(dth12/dthc(i)*thc_num(i))+1);
    for j=2:length(ix_sp)
      th1=th_sp(ix_sp(j-1))+sp_tol;
      th2=th_sp(ix_sp(j))-sp_tol;
      dth12=th2-th1;
      th_tmp=[th_tmp linspace(th1,th2,round(dth12/dthc(i)*thc_num(i)))];
    end
    th1=th_sp(ix_sp(end))+sp_tol;
    th2=thc(i+1);
    dth12=th2-th1;
    th_tmp=[th_tmp linspace(th1,th2,round(dth12/dthc(i)*thc_num(i))+1)];
  end
  th_well=[th_well th_tmp(2:end-1)];
end
if N==2
  th_well=th_well(find(th_well>pi12 & th_well<pi32));
```

```
end
Th_init{1,1}=th_well;
R_init{1,1}=r_well*ones(1,length(Th_init{1,1}));
[x_int{1,1},y_int{1,1}]=fun_int(dlt,R_init{1,1},Th_init{1,1},-1,...
  thc,r_trm(:,1),0,0,0,atol,rtol,k,muRTC,mu,pX,acs0);

[m,n]=size(x_int{1,1});
dth=zeros(l_sp,1);
for i=1:l_sp
  Th_init{2,i}=th_well;
  if N==2
    Th_init{2,i}=Th_init{2,i}(find(Th_init{2,i}>pi12 & Th_init{2,i}<pi32));
  end
  xunit=r_circle*cos(Th_init{2,i})+x_sp(i);
  yunit=r_circle*sin(Th_init{2,i})+y_sp(i);
  th_in=atan2(yunit,xunit);
  r_in=sqrt(xunit.^2+yunit.^2);
  ix_in=find(r_in>r_well);
  R_init{2,i}=r_in(ix_in);
  Th_init{2,i}=th_in(ix_in);

  % pick 2 lines issuing from the well closest to s.p angle and
  % ending on perimeter (one on each side)
  ix_sp_ray=find(sign(th_well-th_sp(i))>0);
  if ~isempty(ix_sp_ray)
    ix_sp_p=ix_sp_ray(1);
    ix_sp_m=ix_sp_ray(1)-1;
    if ix_sp_m==0
      ix_sp_m=n;
    end
  else
    ix_sp_p=1;
    ix_sp_m=n;
  end
  th_arc0=atan2(y_int{1,1}(end,ix_sp_m),x_int{1,1}(end,ix_sp_m));
  th_arc0=th_dlm(th_arc0,thc(1),thc(end),2*pi);
  th_arcf=atan2(y_int{1,1}(end,ix_sp_p),x_int{1,1}(end,ix_sp_p));
  th_arcf=th_dlm(th_arcf,thc(1),thc(end),2*pi);
  th_arc=[th_arc0 th_arcf];
  if N==2
    th_arc(2)=pi12*(y_sp(i)>0)+pi32*(y_sp(i)<=0);
  end
  th_arc=sort(th_arc);
  if ~(th_sp(i)-th_arc(1)>=0 && th_arc(2)-th_sp(i)>=0)
    th_arc(1)=th_arc(1)+2*pi;
    th_arc=sort(th_arc);
  end
  if ~(th_sp(i)-th_arc(1)>=0 && th_arc(2)-th_sp(i)>=0)
    th_sp(i)=th_sp(i)+2*pi;
  end
  Th_init{3,i}=th_arc;
  dth(i)=diff(th_arc);
end

dthf=dth/(2*pi);
thc_num=round(dthf*num);
ix_num=find(thc_num<num_min);
thc_num(ix_num)=num_min;
```

```
for i=1:l_sp
  dth1=th_sp(i)-Th_init{3,i}(1);
  dth2=Th_init{3,i}(2)-th_sp(i);
  % the factor of 2 maintains a density similar to the well parcel
  % since half the lines would escape the domain
  num1=2*round(abs(dth1/dth(i))*thc_num(i));
  num1=num1+num_min*(num1==0);
  th_tmp1=linspace(Th_init{3,i}(1),th_sp(i)-sp_tol,num1);
  num2=2*round(abs(dth2/dth(i))*thc_num(i));
  num2=num2+num_min*(num2==0);
  th_tmp2=linspace(th_sp(i)+sp_tol,Th_init{3,i}(2),num2);
  th_tmp=[th_tmp1 th_tmp2];
  if N==2
    th_tmp=th_tmp(find(th_tmp>pi12 & th_tmp<pi32));
  else
    th_tmp=th_tmp(2:end-1);
  end
  Th_init{3,i}=th_tmp;
  R_init{3,i}=h*ones(1,length(Th_init{3,i}));
end

for j=1:l_sp
  [x_int{2,j},y_int{2,j}]=fun_int(dlt,R_init{2,j},Th_init{2,j},-1,...
    thc,r_trm(:,2),x_sp(j),y_sp(j),r_circle,atol,rtol,k,muRTC,mu,pX,acs0);
end

for j=1:l_sp
  [x_int{3,j},y_int{3,j}]=fun_int(dlt,R_init{3,j},Th_init{3,j},-1,...
    thc,r_trm(:,3),0,0,h,atol,rtol,k,muRTC,mu,pX,acs0);
```

```
end

gap_flag=1;
if gap_flag
  x_prm=x_int{1,1}(end,:);
  y_prm=y_int{1,1}(end,:);
  for j=1:l_sp
    x_prm=[x_prm x_int{3,j}(1,:) x_int{3,j}(end,:)];
    y_prm=[y_prm y_int{3,j}(1,:) y_int{3,j}(end,:)];
  end
  th_prm=atan2(y_prm,x_prm);
  th_prm_dlm=0*th_prm;
  for i=1:length(th_prm)
    th_prm_dlm(i)=th_dlm(th_prm(i),thc(1),thc(end),2*pi);
  end
  th_prm=sort(th_prm_dlm);
  th_prm=[th_prm th_prm(1)+2*pi];
  dth_prm=[diff(th_prm)];
  dth_mean=mean(dth_prm);
  ix_prm=find(dth_prm>3*dth_mean);
  lix_prm=length(ix_prm);
  for i=1:lix_prm
    num_dth=round((th_prm(ix_prm(i)+1)-th_prm(ix_prm(i)))/dth_mean)+2;
    if num_dth>2
      Th_init{4,i}=linspace(th_prm(ix_prm(i)),th_prm(ix_prm(i)+1),num_dth);
      Th_init{4,i}=Th_init{4,i}(2:end-1);
      R_init{4,i}=Th_init{4,i}*0+h;
      [x_int{4,i},y_int{4,i}]=fun_int(dlt,R_init{4,i},Th_init{4,i},1,...
       thc,r_trm(:,4),0,0,h,atol,rtol,k,muRTC,mu,pX,acs0);
    end
  end
else
  lix_prm=0;
end

% post-mortem halt & extrapolation
for i=3:4
  for j=1:(l_sp*(i<4)+lix_prm*(i==4))
    [mp,np]=size(x_int{i,j});
    for n=1:np
      [x_int{i,j}(:,n),y_int{i,j}(:,n)]=...
       exterp_sl(x_int{i,j}(:,n),y_int{i,j}(:,n),h);
    end
    [ixi,ixj]=find(isnan(x_int{i,j}));
    x_int{i,j}(:,ixj)=[];
    y_int{i,j}(:,ixj)=[];
  end
end

% remove lines shorter than ell
ell=0.01*h;
for i=2:(3+(lix_prm>0))
  for j=1:(l_sp*(i<4)+lix_prm*(i==4))
    r=(x_int{i,j}(end,:)-x_int{i,j}(1,:)).^2+...
      (y_int{i,j}(end,:)-y_int{i,j}(1,:)).^2;
    r=sqrt(r);
    ix=find(r>ell);
    x_int{i,j}=x_int{i,j}(:,ix);
    y_int{i,j}=y_int{i,j}(:,ix);
  end
end

% streamline intersection check
[mp,np]=size(x_int);
X_int=[]; Y_int=[];
for i=1:mp
  for j=1:np
    X_int=[X_int x_int{i,j}];
    Y_int=[Y_int y_int{i,j}];
  end
end
[mp,np]=size(X_int);
stop=0;
for i=1:np
  for j=(i+1):np
    cl=(X_int(1,i)>=X_int(1,j));
    xl=X_int(1,i)*cl+X_int(1,j)*(~cl);
    cr=(X_int(end,i)<=X_int(end,j));
    xr=X_int(end,i)*cr+X_int(end,j)*(~cr);
    if xl>xr, continue; end
    if xl==xr
    else
      xlr=linspace(xl,xr,mp);
      yi=interp1(X_int(:,i),Y_int(:,i),xlr);
      yj=interp1(X_int(:,j),Y_int(:,j),xlr);
      dy=yi-yj;
      if dy(1)*dy(end)<0
```

```
        report_str=’Notice: streamlines might intersect.’
        stop=1;
        break
      end
    end
  end
  if stop, break; end
end
```

C.5 Function

This function calls ode45, Octave’s main integration facility.

```
function [x,y]=fun_int(dlt,r_in,th_in,int_sign,thc,r_trm,x0,y0,r0,...
 atol,rtol,k,muRTC,mu,pX,acs0)

x=[];
y=[];
n=150;
dlt=[0 dlt];
dlt0=[0 0.01];

for i=1:length(r_in)
  rth_init=[r_in(i) th_in(i)];
  opt=odeset(’events’,@(t,rth) ...
   psle(t,rth,r_trm,x0,y0,-int_sign),’AbsTol’,atol,’RelTol’,rtol);
  if r0>0
    [t,rth,te,rthe,ie]=ode45(@(t,rth) ...
     pslf(t,rth,thc,-1,k,muRTC,mu,pX,acs0),dlt0,rth_init,opt);
    x_tmp=rth(:,1).*cos(rth(:,2));
    y_tmp=rth(:,1).*sin(rth(:,2));
    r_tmp=sqrt((x_tmp(2)-x0)^2+(y_tmp(2)-y0)^2);
    if r_tmp/r0>1
      if int_sign==-1
        continue
      end
    end
  end
  [t,rth,te,rthe,ie]=ode45(@(t,rth) ...
   pslf(t,rth,thc,int_sign,k,muRTC,mu,pX,acs0),dlt,rth_init,opt);
  t1=linspace(t(1),t(end),n)';
  x1=rth(:,1).*cos(rth(:,2));
  y1=rth(:,1).*sin(rth(:,2));
  if int_sign==-1
    x=[x interp1(t,x1,t1)];
    y=[y interp1(t,y1,t1)];
  else
    x=[x flipud(interp1(t,x1,t1))];
    y=[y flipud(interp1(t,y1,t1))];
  end
end
```

C.6 Main Running Files of Arbitrary Sectors

Main running file.

```
clear all
close all
warning off

flag=2;
%flag=0 would integrate and plot, flag=1 would integrate and save, flag=2 will retrie
filepath='RunFiles/';
Input=load('input.txt');
n=size(Input,1);

%nix=1:n;
nix=sort([1:3:n 2:3:n]);
%nix=sort([1:3:n]);

ix_k=perms(1:3);

for i_fig=nix
%try
  i_fig
  tic
  prm=Input(i_fig,:);
  p_out0=prm(1);
  h=prm(2);
  C_init=prm(3);
  k0=prm(4:6);
  thc0=prm(7:8);

  k0_str=[num2str(-log10(k0(1))) '_' num2str(-log10(k0(2))) '_' num2str(-log10(k0(3))
  for j_fig=1:6 %change numbers to plot desired figure  (ex j=2:2)
    k=k0(ix_k(j_fig,:));

    prC_str=[num2str(p_out0) ’_’ num2str(h) ’_’ num2str(C_init)];
    k_str=[num2str(-log10(k(1))) ’_’ num2str(-log10(k(2))) ’_’ num2str(-log10(k(3)))]
    th_str=[num2str(prm(7)) ’_’  num2str(prm(8))];
    file_str=[prC_str ’_’ k_str ’_’ th_str ’_j’ num2str(j_fig) ’.mat’];
    folder_str=[filepath th_str ’/’ k0_str ’/’ k_str ’/’];

    prm(4:6)=k;
    const_read
    input_read
    num=45;
    dlt=3000;
    atol=1e-12;
    rtol=1e-10;
    r_circle=h/7.5;
    r_well=h/20;

    if flag==0
      psl
      psl_plot
    elseif flag==1
      psl
      save(file_str,’thc’,’N’,’h’,’r_well’,’x_int’,’y_int’)
      mkdir(folder_str);
      system([’mv ’ file_str ’ ’ folder_str]);
    elseif flag==2
      load([folder_str file_str])
```

```
      psl_plot
    end

  end



  toc
%catch
  display(['config ' num2str(prm) ' skipped'])
  continue
%end

end

%p2sl_surf_sim(th0,prm,2)
```

C.7 Post-Integration Fixes

This function attempts to extrapolate pathlines in the case of halt failure.

```
function [x,y]=exterp_sl(x,y,r0)

m=length(x);
r=sqrt(x.^2+y.^2)/r0;
ix=find(r>1);
if isempty(ix)
  return
end
```

```
ell=1e-3;
ix=find(r<1-ell);
lix=length(ix);
if lix<2
  x=x*NaN;
  y=y*NaN;
  return
end
x=x(ix);
y=y(ix);
m1=ix(1)-1;
m2=m-ix(end);
if m1>0
  dl=linspace(ell,1.5*r0,2*m1)'; %(ell:0.01*r0:3*r0)';
  ldl=length(dl);
  dx=x(1)-x(2);
  dy=y(1)-y(2);
  th_end=atan2(dy,dx);
  x_ex=x(1)+dl*cos(th_end);
  y_ex=y(1)+dl*sin(th_end);
  r_ex=sqrt(x_ex.^2+y_ex.^2)/r0;
  if r_ex(end)<1
    x_ex=x_ex(1:m1);
    y_ex=y_ex(1:m1);
  else
    i1=interp1(r_ex,1:ldl,1);
    ix=find(r_ex<1);
    x_ex=[x_ex(ix); interp1(1:ldl,x_ex,i1)];
    y_ex=[y_ex(ix); interp1(1:ldl,y_ex,i1)];
    x_exi=linspace(x_ex(1),x_ex(end),m1)';
    y_ex=interp1(x_ex,y_ex,x_exi);
    x_ex=x_exi;
  end
  x=[x_ex; x];
  y=[y_ex; y];
end
if m2>0
  dl=linspace(ell,1.5*r0,2*m2)';
  ldl=length(dl);
  dx=x(end)-x(end-1);
  dy=y(end)-y(end-1);
  th_end=atan2(dy,dx);
  x_ex=x(end)+dl*cos(th_end);
  y_ex=y(end)+dl*sin(th_end);
  r_ex=sqrt(x_ex.^2+y_ex.^2)/r0;
  if r_ex(end)<1
    x_ex=x_ex(end-m2+1:end);
    y_ex=y_ex(end-m2+1:end);
  else
    i2=interp1(r_ex,1:ldl,1);
    ix=find(r_ex<1);
    x_ex=[x_ex(ix); interp1(1:ldl,x_ex,i2)];
    y_ex=[y_ex(ix); interp1(1:ldl,y_ex,i2)];
    x_exi=linspace(x_ex(1),x_ex(end),m2)';
    y_ex=interp1(x_ex,y_ex,x_exi);
    x_ex=x_exi;
  end
  x=[x; x_ex];
  y=[y; y_ex];
end
```

C.8 Plotting Arbitrary Sectors

This file plots visuals from saved integration files.

```
figure(i_fig)
set(i_fig,'position',[552 100 700 714])
hax(i_fig,j_fig)=subplot(3,2,j_fig);

well_clr=[0.2 0.75 0.5];
sp_clr=[0 0.45 0.74];
arc_clr=[0.63 0.16 0.25];

for i=1:N
  h_th=plot(hax(i_fig,j_fig), cos(thc(i))*[r_well h],sin(thc(i))*[r_well h],'k');
  hold on
end
th=linspace(0,2,500)*pi;
plot(hax(i_fig,j_fig), h*cos(th),h*sin(th),'k')

[mp,np]=size(x_int);
X_int=[]; Y_int=[];
for i=1:mp
  for j=1:np
    X_int=[X_int x_int{i,j}];
    Y_int=[Y_int y_int{i,j}];
  end
end

R_int=sqrt(X_int([1 end],:).^2+Y_int([1 end],:).^2);
R_int=sort(R_int);
h_tol=0.1;
ix_well=find(abs(R_int(1,:)-r_well)/h<h_tol & abs(R_int(2,:)/h-1)<h_tol);
ix_arc=find(abs(R_int(1,:)/h-1)<h_tol & abs(R_int(2,:)/h-1)<h_tol);
if N>2
  h_sl=plot(hax(i_fig,j_fig), X_int,Y_int);
  h_sl0=plot(hax(i_fig,j_fig), X_int(end,ix_arc),Y_int(end,ix_arc),’.’);
else
  h_sl=plot(hax(i_fig,j_fig), X_int,Y_int,-X_int,Y_int);
  h_sl0=plot(hax(i_fig,j_fig), X_int(end,ix_arc),Y_int(end,ix_arc),’.’,...
             -X_int(end,ix_arc),Y_int(end,ix_arc),’.’);
end
set(h_sl,’color’,sp_clr)
set(h_sl(ix_well),’color’,well_clr)
set(h_sl(ix_arc),’color’,arc_clr)
set(h_sl0,’color’,arc_clr,’markersize’,6)   %size of dot
set(h_sl, ’linewidth’, 0.5)

i0=round(0.8*size(X_int,1));
lix_arc=length(ix_arc);
if lix_arc>0
   dx=diff(X_int(i0-(0:1),ix_arc));
  dy=diff(Y_int(i0-(0:1),ix_arc));
  th_arrow=-atan2(dy,dx);
  cth=cos(th_arrow);
  sth=sin(th_arrow);
  wx=0.4; wy=0.2;
  arrow=[1 0 1  wx 1 wx 1;
         0 0 0 -wy 0 wy 0];
  for i=1:length(th_arrow)
    A=[cth(i) sth(i); -sth(i) cth(i)];
    a=0.05*h*A*arrow+[X_int(i0,ix_arc(i)); Y_int(i0,ix_arc(i))];
    h_arr(i)=plot(hax(i_fig,j_fig), a(1,:),a(2,:),'k');
  end
  set(h_arr,'color',arc_clr)
end

axis equal

title(hax(i_fig,j_fig), num2str(k))
set(hax(i_fig,j_fig),'parent',i_fig)
drawnow
```

C.9 Stagnation Points

This function locates stagnation points in the case of three or more sectors.

```
function [th_sp_keep,r_sp_keep]=sploc(thc,k,C,acs0)

a20=acs0(2);
a=acs0(3:2:end);
b=acs0(4:2:end);
```

```
th_sp_keep=[]; r_sp_keep=[];
N=length(k);
for i=1:N
  i1=(i-1)*(i>1)+N*(i==1);
  i2=i*(i>1)+(N+1)*(i==1);
  th_sp=th_dlm(atan2(a(i),b(i))/2,thc(i1),thc(i2),pi/2);
  r_sp=a(i)*sin(2*th_sp)+b(i)*cos(2*th_sp)-1;
  r_sp=-a20*k(i)./(C*r_sp);
  ix=find(~isnan(th_sp) & r_sp>0);
  th_sp_keep=[th_sp_keep th_sp(ix)];
  r_sp_keep=[r_sp_keep sqrt(r_sp(ix))];
end
th_sp_keep=th_sp_keep(:);
r_sp_keep=r_sp_keep(:);
```

C.10 Delimiting the Azimuthal Angle

This function rewinds the azimuthal angle to find the correct position in the circle.

```
function th=th_dlm(th,th1,th2,skip)

if th<=th1
  i=ceil((th1-th)/skip);
  j=floor((th2-th)/skip);
  th=th+(i:j)*skip;
elseif th>=th2
  i=ceil((th-th2)/skip);
  j=floor((th-th1)/skip);
  th=th-(i:j)*skip;
else
  i=floor((th-th1)/skip);
  j=floor((th2-th)/skip);
  th=th+(-i:j)*skip;
end
if length(th)>1
  tol=1e-15;
  n=(th(end)-th(1))/(2*pi);
  if (abs(rem(n,1))<tol || abs(rem(n,1)-1)<tol)
    th=th(1:end-1);
  end
end
```