Characterization of porous filtration media

Industrial Representative: U. Beuscher (W. L. Gore)

Participants: M. Davis (CGU), J. Dehner (CGU), J.D. Fehribach (WPI), P.-W. Fok (Cal Tech), Q. Han (Delaware), D. Herbst (Chicago), J. Jagalur-Mohan (RPI), K. Newhall (RPI), V. Peridier (Temple), L. Rossi (Delaware), J. Siddique (Penn State, York), A.C. Slim, S. Swaminathan (Northwestern), Y. Yang (LSU)

Final presentation: M. Davis, 19 June 2009.

Report: J. Dehner, J.D. Fehribach, D. Herbst, K. Newhall, V. Peridier, A.C. Slim (editor), S. Swaminathan, Y. Yang.

Abstract We present a model and numerical code for an idealized 2D porous medium consisting of a rectangular lattice of cylindrical pores of equal length and pore diameters taken from a prescribed distribution. We calculate the flow field through the medium and its permeability and tortuosity. We also present simulations of porosimetry, porometry and particle-filtration experiments. For a single layer of pores we complement these simulations with analytic predictions for the expected observations. We also discuss extensions of these analytic predictions to multi-layer media for porosimetry experiments at a small pressure drop and filtration when only a few pores have been blocked.

1 Introduction

Filters are widely used in many industrial as well as electronic processes, thus there is a need to understand the fundamentals of their performance. In particular, to identify defective filters during manufacturing, it is useful to be able to characterize their properties in a non-destructive way. In this report our working group considered the mathematical modeling and characterization of an idealized 2D porous medium and simulated its use as a particle filter. The characteristics we considered are the resistance to fluid flow, the pore-size distribution and the filtration efficiency.

Simple, single measurements representative of the resistance to fluid flow are the permeability and tortuosity. These are, respectively, the effective resistance of the porous medium to fluid flow through it and the expected length of a fluid path through the medium relative to its thickness.

The pore-size distribution can be obtained via several techniques. In porosimetry the sample is underlain by a low permeability substrate and saturated with a wetting fluid of known surface tension. Gas is introduced at the top of the sample and its pressure progressively increased. As the applied pressure is ramped up, pores at the exposed surface of the evaluation material can evacuate — and simultaneously adopt the applied pressure. The displaced liquid is ejected from the base of the substrate. The liquid content of pores directly connected to already-vacated pores may also evacuate. As the pressure increases the number of open pores and length of linked chains of vacated pores grows. (Note that the pressure in the low-porosity substrate remains at zero. These pores are never evacuated, because their pore diameters are tiny.) The measurement obtained from such a porosimetry experiment is the curve of volume of drained liquid against applied pressure. Techniques exist to infer a pore-size distribution from this data. Figure 1 illustrates the experimental process.

A second, similar, technique is porometry. The sample is again saturated with a wetting liquid and gas is introduced at the top of the sample; no low-permeability substrate underlies the sample. The pressure at

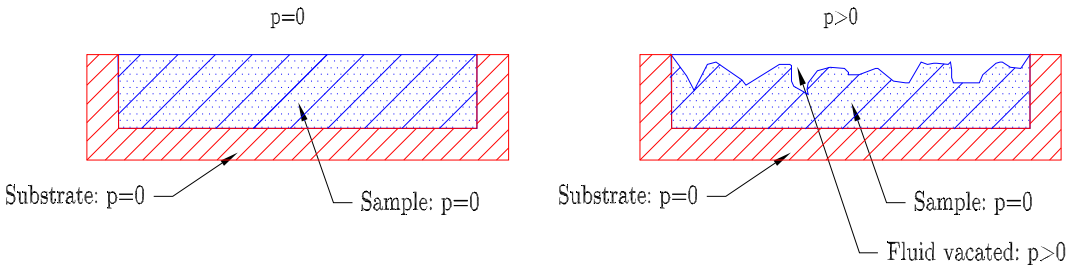

FIGURE 1: *Schematic of a porosimetry experiment. Left: start of experiment. Right: after an increase in applied pressure.*

the top is again gradually increased and progressively pores drain. At a critical pressure, the bubble point, a completely drained path appears through the filter and gas begins to flow continuously through it. With increasing applied pressure, the flux increases and more paths open. The desired data is the gas flux through the sample against applied pressure, from which a pore-size distribution can also be inferred.

Ultimately, destructive testing is also desirable in characterizing how a particular porous medium performs as a filter. Most important is the minimum particle size that will be captured. This can be described by the probability that a particle of a given size will be captured by a new filter. As smaller pores get clogged with particles, fewer and fewer pores remain to trap new particles. At some point all small pores are clogged and all particles pass through the filter unobstructed. Two additional relevant quantities are thus the efficiency of the filter with age (the probability of capturing a particle after multiple particles have entered the filter) and the expected lifetime (the time until the filter is full and unable to capture any new particles).

Our aim is to simulate these three different processes: porosimetry, porometry, and filtration, for an idealized 2D porous medium. This idealized mediums consists of a rectangular lattice of pipes of the same length and diameters taken from a specified distribution. The specific questions to be addressed are:

- How is the flow distributed through the structure? What is the permeability and tortuosity?
- What is the observed pore-size distribution of the structure measured by the various characterization techniques (porosimetry, porometry)? What is the bubble point? The motivation here is that simulations for a medium of known pore-size distribution, and subsequent inversion of the experimental data by existing techniques to obtain an inferred pore-size distribution, will highlight the accuracy of the inversion technique, and provide insight into potential improvements.
- What is the filtration efficiency (probability of capture) of the network for a certain particle size?
- How do permeability and filtration efficiency change over time with the deposition of particles?

The report is structured as follows. In §2 we describe the idealized porous medium in more detail. We introduce the relevant variables and processes and non-dimensionalize them. Substantial progress and useful insights are possible for a single layer of pores where we can immediately predict the expected measurements from the underlying pore-size distribution. We describe the relevant theory for porosimetry and filtration in §3. In §4 we turn to multi-layer porous media. We describe how to calculate the flow field for a given realization of the lattice in §§4.1 and 4.2, and give expressions for the permeability and tortuosity in §4.4. We simulate porosimetry in §4.5, porometry in §4.6, and filtration in §4.7. We predominantly consider the forward calculation, simulating the various experimental outputs for the idealized medium generated, and leave the inversion of the data for the analyses used already for real experiments. However for the porosimetry experiment we also give a discussion of the comparison between the original and inferred pore-size distributions. We summarize in §5.

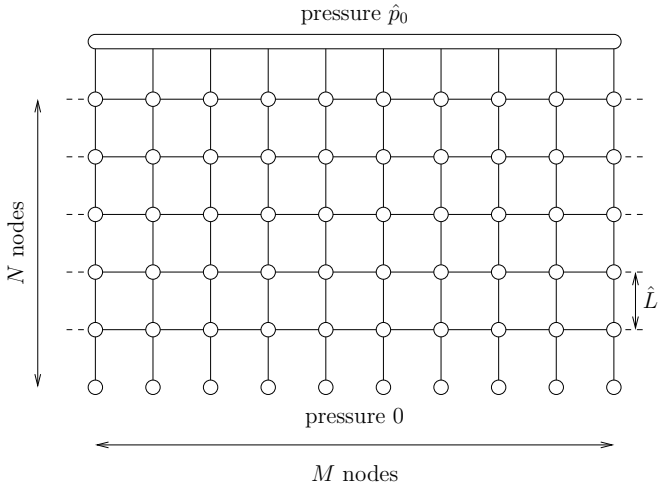

FIGURE 2: *Porous medium of cylindrical capillaries. The open volume at the top is combined into a single node. The network can be enlarged by imposing periodic boundaries (dashed edges).*

2 Lattice model

We consider an $N \times M$ rectangular lattice network of cylindrical capillaries as an idealized porous medium, as shown in figure 2. We assume the pore volume is mostly in the edges, with a negligible volume in the nodes, and thus use the terms edges and pores interchangeably in the remainder of the report.* For later convenience we combine the open volume at the top of the medium as a single node. We term the bottom row of nodes the sink nodes, and the top node as the source node. Note that in this framework, a single-layer porous medium is a $1 \times M$ lattice.

In practice a filter is thin ($M \gg N$), but this imbalanced lattice aspect ratio is computationally undesirable. Instead, given N we can select a computationally reasonable number for the M columns and then use periodic boundary conditions. Periodic boundary conditions conceptually connect infinitely many copies of the $N \times M$ lattice at the lateral sides, or equivalently we can add $N - 1$ horizontal edges, each edge extending from the last node in a given row to the first node in that same row as depicted by the dashed lines in figure 2.

Each edge i models a cylinder of equal length $\hat{L}$ and diameter $\hat{d}_i$ taken from a specified probability distribution. The diameters are chosen independently. Volumes of each edge are $\hat{V}_i = \pi \hat{L} \hat{d}_i^2/4$. We use hats to denote dimensional variables.

In flow through a dry network or through a partially wetted network in a porometry experiment, a pressure $\hat{p}_0$ is applied at the top of the lattice with zero pressure at the bottom. This generates a pressure drop $\Delta\hat{p}_i$ across each edge i and drives a flux of gas through that edge according to the Hagen–Poiseuille law

$$\hat{Q}_i = -\frac{\pi \hat{d}_i^4}{128 \mu \hat{L}} \Delta \hat{p}_i,$$

where μ is the dynamic viscosity of the fluid.

During a porosimetry experiment, the lattice is partly filled with a wetting liquid at gauge pressure and partly with a pressurized gas. If we consider a single liquid-filled pore i that is exposed to the gas at one end, then there exists a critical gas pressure $\hat{p}_i^c$ at which the pore drains and the intruding gas displaces the resident liquid. This critical gas pressure is the capillary pressure for the pore given by $\hat{p}_i^c = 4\gamma \cos\theta/\hat{d}_i$,

*For most porous media this will not be a good approximation, however our approach could be extended to more complex representations, in particular ones having a volume associated with each node.

where γ is the surface tension coefficient of a liquid–gas interface and θ the contact angle of such an interface with the solid cylinder walls.

2.1 Non-dimensionalization

We non-dimensionalize variables as follows: we scale edge diameters by the mean edge diameter $\langle d \rangle$, lengths by $\hat{L}$, volumes by $\pi \hat{L} \langle d \rangle^2 / 4$, pressures by $4\gamma \cos\theta / \langle d \rangle$ and fluxes by $\pi \langle d \rangle^3 \gamma \cos\theta / 32 \mu \hat{L}$. Thus in our non-dimensional framework (with non-dimensionalized variables unhatted) the edges have length unity, the diameters d_i are chosen from distributions with mean 1, the volumes $V_i = d_i^2$, the capillary pressures are $p_i^c = 1/d_i$, and the Hagen–Poiseuille law becomes

$$Q_i = -d_i^4 \Delta p_i. \tag{1}$$

A non-dimensional pressure $p_0 = \hat{p}_0 \langle d \rangle / 4\gamma \cos\theta$ is applied at the top of the lattice.

3 Single-layer theory

Rather than start with the full network, we begin by taking $N = 1$ and consider a single layer of M pores. In this special case we can predict the expected observations in the porosimetry experiment, filter efficiency over time, and average lifetime of a filter directly from the pore-diameter distribution.

3.1 Porosimetry

For a single layer of pores, we can analytically compute the result of the porosimetry experiment by obtaining the expectation of the total drained volume V versus applied pressure p_0 in terms of the pore-diameter distribution. As the applied pressure is increased, the total volume of liquid drained from the filter is equal to the volume of all pores with capillary pressure less than the applied pressure difference.

The average volume drained, $\langle V(p_0) \rangle$, is calculated by conditioning on the number of pores that have drained at the applied pressure, p_0:

$$\langle V(p_0) \rangle = \sum_{n=0}^{M} \langle V(p_0) | n \text{ pores drained} \rangle \, P(n \text{ pores drained}), \tag{2}$$

where $\langle \cdot \rangle$ denotes the expected value and $\langle V(p_0) | n \text{ pores drained} \rangle$ is the expected volume of drained pores at pressure p_0 given that exactly n nodes have drained. The volume drained from exactly n open pores is the sum over each individual volume drained. These are independent and identically distributed and thus

$$\langle V(p_0) | n \text{ pores drained} \rangle = n \, \langle V_i(p_0) | i\text{th pore drained} \rangle \tag{3}$$

where V_i is the volume from any one pore. Note that this is not the volume obtained from the average diameter because there is the additional information that the pore is open, or equivalently that $d_i \geq 1/p_0$. Substituting (3) into (2), we find

$$\begin{aligned} \langle V(p_0) \rangle &= \langle V_i(p_0) | i\text{th pore drained} \rangle \sum_{n=0}^{M} n P(n \text{ pores drained}) \\ &= \langle V_i(p_0) | i\text{th pore drained} \rangle \, \langle \text{number of pores drained} \rangle \, , \end{aligned} \tag{4}$$

in other words the expected drained volume is the product of the expected volume of a drained pore and the expected number of drained pores.

The probability that a randomly selected pore is open is $1 - F_D(1/p_0)$, where $F_D(1/p_0) = \int_0^{1/p_0} f_D(d) \, \mathrm{d}d$, with $f_D(d)$ the pore-diameter probability density function (PDF). Thus the expected number of open pores is $M(1 - F_D)$.

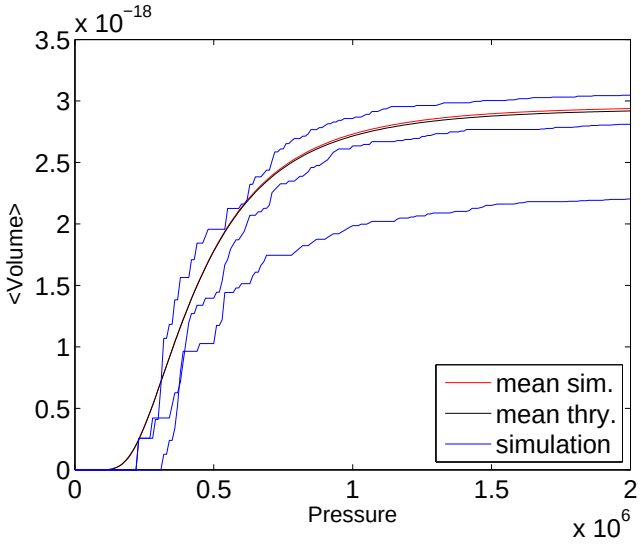

FIGURE 3: *Volume drained from single-layer filter as a function of applied pressure. Shown in blue are three of* 5000 *numerical experiments, in red the average of these numerical experiments and in green, the computation from (4). The* 100 *pore diameters were gamma distributed with parameters* $\alpha = 2$ *and* $\beta = 0.5 \times 10^{-6}$.

The expected volume of a drained pore is

$$\langle V_i(p_0) | i\text{th pore drained} \rangle = \int_{1/p_0}^{\infty} d^2 f_D(d)\, \mathrm{d}d. \tag{5}$$

We now have both components required to evaluate (4). Figure 3 shows the $\langle V \rangle$ versus p_0 curve and various numerical simulations.

3.2 Filter efficiency

We now analyze the filtering efficiency of the single-layer porous medium having M pores. We assume the particles to be filtered are mono-disperse with diameter d_p and that they enter each pore with equal probability. A particle is caught (and blocks the pore) with probability one if it enters a pore having smaller diameter than it and with probability zero if it enters a pore having larger diameter than it. Particles enter the filter one at a time and do not interact with each other.

With this idealized model, we describe the number of pores remaining that can capture a particle (termed "capturing pores" for later convenience) as a Markov Chain. Let the number of capturing pores remaining after n particles have entered the filter be $C^{(n)}$ (the state of the Markov Chain at epoch n) and let $C^{(0)} = c_0$. The probability that a randomly chosen pore is initially a capturing pore is $F_D(d_p)$, the probability the pore has diameter smaller than d_p, where $F_D(d_p) = \int_0^{d_p} f_D(d)\, \mathrm{d}d$. Thus the distribution of the number of initial capturing pores is given by M Bernoulli trials with probability of success f_p,

$$P(C^{(0)} = c_0) = \begin{pmatrix} M \\ c_0 \end{pmatrix} F_D(d_p)^{c_0} [1 - F_D(d_p)]^{M - c_0}. \tag{6}$$

When a particle enters the ideal filter, it has an equal probability of entering each unblocked pore. The probability that it is captured is the probability that it enters one of the remaining capturing pores out of all unblocked pores. If the particle enters a capturing pore, then the number of capturing pores is reduced

by one, otherwise it remains the same. This can be interpreted as the following stochastic update rule:

$$C^{(n+1)} = \begin{cases} C^{(n)} & \text{with probability} \quad 1 - \frac{C^{(n)}}{M-(c_0-C^{(n)})} \\ C^{(n)} - 1 & \text{with probability} \quad \frac{C^{(n)}}{M-(c_0-C^{(n)})} \end{cases} . \tag{7}$$

Note that this update rule depends on the number of capturing pores available initially, c_0.

The expected filter efficiency $\eta^{(n)}$ after n particles have entered the filter is the ratio of the expected number of capturing pores remaining (equivalently the expected state of the Markov chain at epoch n) to the total number of pores:

$$\eta^{(n)} = \frac{1}{M}\left\langle\left\langle C^{(n)}|C^{(0)}=c_0\right\rangle\right\rangle = \frac{1}{M}\sum_{c_0=0}^{M}\left\langle C^{(n)}|C^{(0)}=c_0\right\rangle P(C^{(0)}=c_0). \tag{8}$$

The average lifetime of the filter $\langle T\rangle$ is measured as the expected number of particles to enter the filter before the number of capturing pores reaches zero (equivalently the mean first passage time to zero of the Markov chain):

$$\langle T\rangle = \left\langle\left\langle T|C^{(0)}=c_0\right\rangle\right\rangle, \tag{9}$$

where the quantity $\left\langle T|C^{(0)}=c_0\right\rangle$ is the average number of particles that enter the filter before $C^{(n)}=0$ is first achieved, given that $C^{(0)}=c_0$.

To calculate the quantities $\left\langle C^{(n)}|C^{(0)}=c_0\right\rangle$ and $\left\langle T|C^{(0)}=c_0\right\rangle$, we recast the update rule (7) as the $(c_0+1)\times(c_0+1)$ probability transition matrix

$$\mathsf{P}_C = \begin{pmatrix} 1 & \frac{1}{M-(c_0-1)} & 0 & \dots & 0 & 0 \\ 0 & 1-\frac{1}{M-(c_0-1)} & \frac{2}{M-(c_0-2)} & \dots & 0 & 0 \\ 0 & 0 & 1-\frac{2}{M-(c_0-2)} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-\frac{c_0-1}{M-1} & \frac{c_0}{M} \\ 0 & 0 & 0 & \dots & 0 & 1-\frac{c_0}{M} \end{pmatrix}, \tag{10}$$

where $(P_C)_{ij}$ is the probability of transitioning from $j-1$ capturing pores to $i-1$ capturing pores after one additional particle has entered the filter. Let $\phi^{(n)}$ be the (c_0+1) component vector with entries $\phi_i^{(n)}$ the probability that $C^{(n)}=i-1$. Then the initial condition is given by $\phi^{(0)}=(0,\dots,0,1)^t$ and the probabilities at later times are [1, §2.1]

$$\phi^{(n)} = \mathsf{P}_C^n\phi^{(0)}. \tag{11}$$

The expected number of capturing pores remaining after n particles have entered the filter is then

$$\left\langle C^{(n)}|C^{(0)}=c_0\right\rangle = (0,1,2,\dots,c_0)\cdot\phi^{(n)}. \tag{12}$$

To compute the expected number of particles that enter the filter before no capturing pores remain, we partition the matrix P_C as

$$\mathsf{P}_C = \begin{pmatrix} 1 & \mathbf{S}^t \\ 0 & \\ \vdots & \mathsf{Q} \\ 0 & \end{pmatrix} \tag{13}$$

where $\mathbf{S}^t$ is a row vector and Q a matrix. Then the expected number of particles to enter the filter before all capturing pores are blocked having started with c_0 capturing pores initially, $\left\langle T|C^{(0)}=c_0\right\rangle$, is w_{c_0}, the c_0th component of the vector [1, §3.3]

$$\mathbf{w} = (\mathsf{I}-\mathsf{Q})^{-1}(1,1,\dots,1)^t. \tag{14}$$

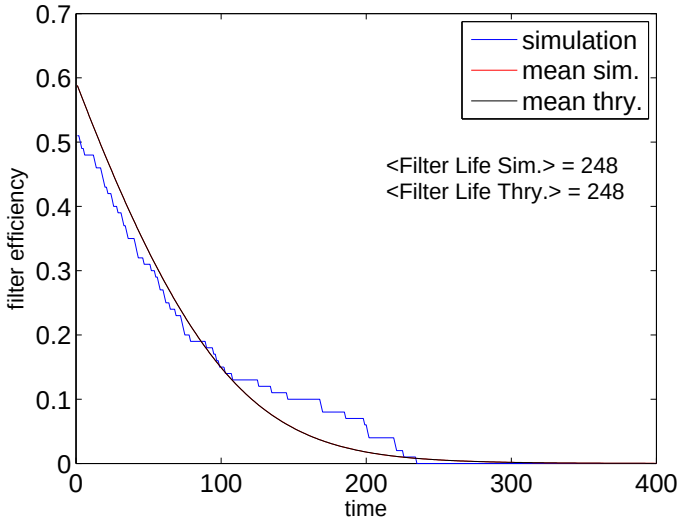

FIGURE 4: *Filter efficiency calculated as the probability of capturing a particle of a given size as a function of number of particles having entered the filter (time). One numerical experiment is shown in blue, the average of* 5000 *numerical experiments is in red and the expected value is in green (the latter two overly one another). The* 100 *pore diameters were gamma distributed with parameters* $\alpha = 2$ *and* $\beta = 0.5 \times 10^{-6}$.

(This can be interpreted as the Markov chain gaining reward 1 for each transient state it visits. Thus the average reward is the average time spent in transient states.)

We now have all the components to evaluate (8) and (9). Figure 4 shows the expected filter efficiency and lifetime together with results from a numerical simulation.

4 Multi-layer simulations

We now turn to lattices consisting of multiple layers, $N > 1$. Here it is no longer apparent that we can calculate the relevant experimental curves directly from the pore-diameter distribution and we resort to numerical simulations, except for porosimetry experiments at a small pressure drop and filtration when only a few pores have been blocked.

4.1 Pressure and flow distributions: matrix approach

To simulate the behaviour of the porous medium in the various processes, we need to calculate the flow fields through the edges and the pressures at the nodes. In most of the report we use a matrix approach for finding these, as described below. This allows calculations to be done cleanly and can be efficient for large systems if sparse-matrix approaches are harnessed. However even sparse-matrix approaches may not be sufficient for very large systems and we also describe an iterative approach for finding the flow field in §4.2.

We represent the network of pores by an incidence matrix. To do so we number the nodes and edges of the network and assign directions to the edges as shown for a typical (small) network in figure 5 (here we use the non-periodic lattice network shown in figure 2). Note that the assigned directions do not impose a flow direction, but only indicate which direction corresponds to a positive flow (or pressure drop).

The entries of the incidence matrix D are given by

$$D_{i\bar{j}} = \begin{cases} -1 & \text{if edge } i \text{ points away from node } \bar{j} \\ +1 & \text{if edge } i \text{ points towards node } \bar{j} \\ 0 & \text{otherwise.} \end{cases}$$

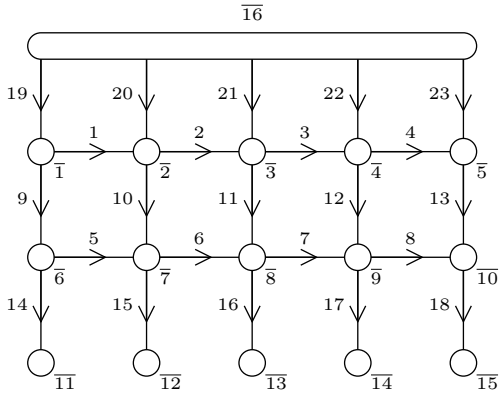

FIGURE 5: *The numbering and direction convention for the network. Edges are numbered horizontally first from top to bottom (top left to bottom right), then vertically top to bottom (top left to bottom right, omitting the top row of edges), then the top row connecting to the top-most, open node from left to right. Nodes are numbered top to bottom from top left to bottom right (omitting the top-most, open node) and then the top, open node. We use over-lined numbers to indicate a node and absence of decoration to indicate an edge.*

For example, the incidence matrix for a 2×2 network is

$$\mathsf{D} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix}.$$

Using the incidence matrix, we calculate the flow field and pressure drops throughout the network as follows. The pressure drop across each edge is given by

$$\Delta\mathbf{p} = \mathsf{D}\mathbf{p}, \tag{15}$$

where $\Delta\mathbf{p}$ is the vector of pressure drops and $\mathbf{p}$ is the vector of pressures at each node. The flow through each edge is connected to the pressure drop across that edge via

$$\mathbf{Q} = \underbrace{\begin{pmatrix} -d_1^4 & 0 & 0 & \cdots & 0 \\ 0 & -d_2^4 & 0 & \cdots & 0 \\ 0 & 0 & -d_3^4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -d_{MN+(M-1)(N-1)}^4 \end{pmatrix}}_{\mathsf{R}^{-1}} \Delta\mathbf{p}, \tag{16}$$

from the Hagen–Poiseuille law (1), where $\mathbf{Q}$ is the vector of volume fluxes through each edge and the matrix R is termed the resistance matrix. Finally, the net volume flux at each node is given by

$$\mathbf{q} = \mathsf{D}^t\mathbf{Q}, \tag{17}$$

where $\mathbf{q}$ is the vector of net fluxes at each node and superscript t denotes transpose. Combining (15)–(17), we have

$$\mathbf{q} = \underbrace{\mathsf{D}^t\mathsf{R}^{-1}\mathsf{D}}_{\mathsf{L}}\mathbf{p}. \tag{18}$$

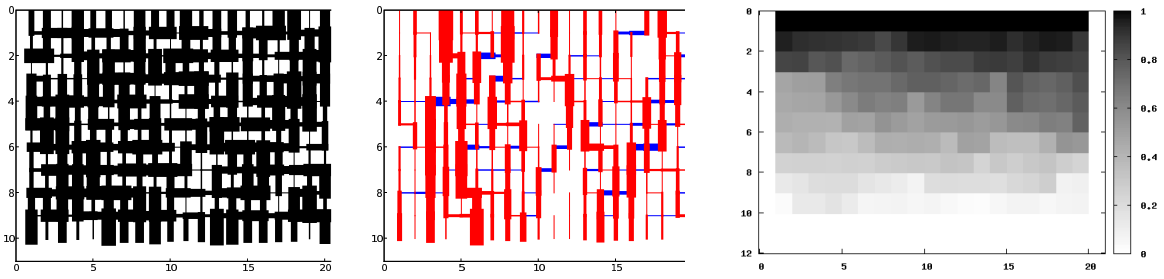

FIGURE 6: *Single realization of a non-periodic* 20×10 *network with diameters taken from a uniform distribution between* 0.5 *and* 1.5. *(a) Pore diameters, with line width reflecting relative diameter. (b) Volume flux through pores for* $p_0 = 1$, *with line width reflecting relative magnitudes of the flux. Red indicates flow with the assigned direction, blue is against. Some pores have such small flux that the discretized line width is zero. (c) Pressures at nodes for* $p_0 = 1$.

Note that the Laplacian matrix L is symmetric.

Continuity requires that the net flux at all interior nodes is zero, while the pressure at the source node is p_0 and the pressure at the sink nodes is zero. Using these conditions, we partition the matrix L to give

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{q}_s \end{pmatrix} = \begin{pmatrix} \mathsf{A} & \mathsf{B}^t \\ \mathsf{B} & \mathsf{C} \end{pmatrix} \begin{pmatrix} \mathbf{p}_i \\ \mathbf{p}_s \end{pmatrix},$$

where $\mathbf{p}_i$ is the vector of pressures at interior nodes, $\mathbf{q}_s$ is the vector of net fluxes at the source and sink nodes, and $\mathbf{p}_s$ is the vector of pressures at the source and sink nodes. Thus we solve

$$\mathbf{0} = \mathsf{A}\mathbf{p}_i + \mathsf{B}^t\mathbf{p}_s, \quad \mathbf{q}_s = \mathsf{B}\mathbf{p}_i + \mathsf{C}\mathbf{p}_s, \tag{19}$$

for the unknowns $\mathbf{p}_i$ and $\mathbf{q}_s$, from which $\mathbf{p}$, $\Delta\mathbf{p}$, $\mathbf{Q}$, and $\mathbf{q}$ are obtained. The flux into the system is $q_{\overline{MN+1}}$ and the flux out of the system is

$$\sum_{\overline{j}:\text{sink nodes}} q_{\overline{j}} = -q_{\overline{MN+1}}.$$

Sample pressure and flow fields are shown in figure 6.

4.2 Pressure and flow distributions: iterative approach

In the previous section we reduced the problem of finding the pressure and flow distributions to a system of $M \cdot N + 1$ linear equations (one equation per node). Even though the system is sparse, for $M \cdot N$ large it may be impractical to solve this system exactly. Therefore, we developed an iterative algorithm to find approximate solutions.

4.2.1 Hardy–Cross algorithm

Before we found a successful algorithm for large systems, we experimented with several variations of the established Hardy–Cross algorithm. This algorithm makes use of the following three conditions on the pressure and flow fields: +

1. The net volume flux at any node must be zero.
2. The pressure at the top is p_0, and the pressure at the bottom is zero.
3. The sum of the pressure drops around the edges of any unit cell is zero: $0 = \sum_{j\in\square} \sigma_j R_{jj} Q_j$, with $\mathbf{Q}$ and R as defined in §4.1, and $\sigma_j = 1$ if edge j is followed in its prescribed direction and $\sigma_j = -1$ otherwise.

The algorithm starts with a guess that satisfies condition 1, and makes small corrections to improve conditions 2 and 3, while guaranteeing that condition 1 stays satisfied. One Hardy–Cross iteration works as follows:

- Choose a unit cell (for example randomly).

- Calculate $\Delta p = \sum_{j\in\square} \sigma_j R_{jj} Q_j$. In general $\Delta p \neq 0$.
- Calculate the necessary corrective flow, $\Delta Q = -\Delta p / \sum_{j\in\square} R_{jj}$.
- Add $\sigma_j \Delta Q$ to each edge in the unit cell.

Unfortunately, we found that this algorithm does not converge well for large systems: every time a correction is added to a certain cell to cancel a circulation, the four neighboring cells are corrupted with new circulation. Therefore, the algorithm tends to move the circulation around instead of directly cancelling it. It only works well when the circulation can collect in a cell with large total resistance $\sum_{j\in\square} R_{jj}$, so that it can be cancelled with a small corrective flow. We believe that it would be difficult to implement a scheme of choosing cells correctly so as to guarantee convergence. Therefore, we developed a new method for solving the problem.

4.2.2 Iterations on pressure

After abandoning the Hardy–Cross algorithm we found a different approach that works well for large systems. Instead of beginning with a solution that satisfies condition 1 and iterating condition 3, we do the reverse. We start with a guess for the pressure distribution on the nodes that satisfies condition 2. Then the flow distribution is *defined* to satisfy condition 3:

$$Q_k = \frac{p_{\bar{j}} - p_{\bar{i}}}{R_{kk}}, \tag{20}$$

where edge k goes from node $\bar{i}$ to node $\bar{j}$. These derived flow rates do not immediately satisfy volume conservation, condition 1, but we use an iterative algorithm to converge to a solution satisfying condition 1 to within a specified tolerance.

First, we allow fluid to build up at each node in an imaginary reservoir. We imagine that the amount of fluid in each reservoir determines its pressure linearly by $p_{\bar{i}} = V_{\bar{i}}$. To perform an iteration, we choose a time-step Δt, and for each edge k in the system, we adjust the volume of fluid in nodes $\bar{i}$ and $\bar{j}$ according to the following:

$$V_{\bar{i}}(t+\Delta t) = V_{\bar{i}}(t) - Q_k \Delta t, \qquad V_{\bar{j}}(t+\Delta t) = V_{\bar{j}}(t) + Q_k \Delta t,$$

which is equivalent to

$$p_{\bar{i}}(t+\Delta t) = p_{\bar{i}}(t) - Q_k \Delta t, \qquad p_{\bar{j}}(t+\Delta t) = p_{\bar{j}}(t) + Q_k \Delta t. \tag{21}$$

This correction effectively creates flows from nodes where there is a positive volume flux to nodes where there is a negative flux, which for properly chosen Δt converges to satisfying condition 1.

In addition, before updating each edge in the system according to (21), we set the pressure of the nodes at the top to p_0, and the pressure of the nodes at the bottom to zero, which enforces condition 2. For numerical stability reasons, we add an extra condition that $0 \leq p_{\bar{i}} \leq p_0$ for any node $\bar{i}$ in the system.

To prevent sloshing, where fluid moves back and forth between two adjacent cells on alternating iterations, we must choose a proper time step Δt. First, one can calculate Q_k for every edge in the system using (20). To prevent sloshing, we need to choose Δt so that $Q_k \Delta t$ does not *by itself* cause a change in sign in $(p_{\bar{i}} - p_{\bar{j}})$. (Note, however, that the sign may change do to other connections.) Therefore, we must have $|Q_k \Delta t| \leq \left| \left(p_{\bar{i}} - p_{\bar{j}} \right) / 2 \right|$. Then applying (20) on the second term, we get

$$\Delta t \leq \frac{R_{kk}}{2}.$$

For this condition to apply to the entire system, we simply have

$$\Delta t \leq \frac{\min \mathsf{R}}{2}.$$

In practice, Δt can be scaled down heuristically as the system gets closer to convergence. When the system reaches steady-state, the system has converged to the desired solution.

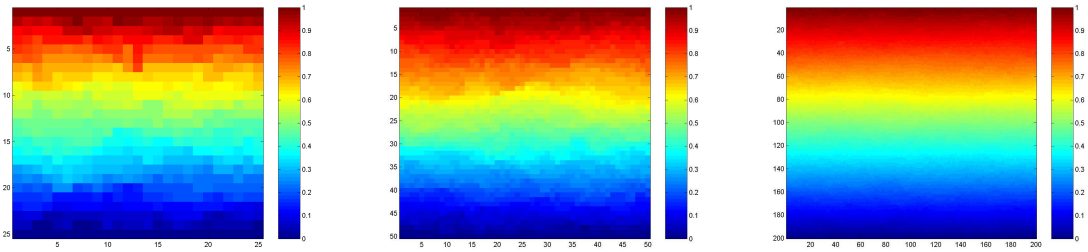

FIGURE 7: *Iterative solution of the pressure distribution with (a)* $M = N = 25$*, (b)* $M = N = 50$*, and (c)* $M = N = 200$*;* $p_0 = 1$*. Pore diameters are gamma distributed with* $k = 9$*,* $\theta = 0.5$*. Periodic boundary conditions connect the left and right sides. Iterations were performed until the convergence criterion,* $\max \Delta p/(p_0/M) < 0.1$*.*

The main advantage of this approach is that, if one starts with a fairly good guess, it may not take many iterations to arrive at an answer within acceptable tolerance. This is especially useful when simulating the motion of individual particles flowing through the filter: when a particle gets trapped and blocks a pore, the new solution will generally be close to the old one.

Sample pressure fields calculated with this algorithm are shown in figure 7. For small systems, the solution has a noticeable wavy appearance, potentially driving significant lateral flow. For larger systems, this pattern becomes only a small perturbation to the dominant linear profile.

4.3 Probabilities and the transition matrix

For finding the tortuosity of the network and for simulating filtration behaviour, we need the probability that a fluid parcel or a particle follows a given edge when at a given node in the lattice. The probability $P_{\overline{ij}}$ that a fluid particle at node $\overline{j}$ arrives at node $\overline{i}$ following a single edge is

$$P_{\overline{ij}} = \frac{\text{outward flux from node } \overline{j} \text{ to node } \overline{i}}{\text{total outward flux from node } \overline{j}}.$$

These probabilities combine into the transition matrix P.

To find P in a compact way using the matrix approach, we use the matrices

$$\mathsf{F} = \begin{pmatrix} Q_1 & 0 & 0 & \cdots & 0 \\ 0 & Q_2 & 0 & \cdots & 0 \\ 0 & 0 & Q_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q_{MN+(M-1)(N-1)} \end{pmatrix} \mathsf{D}$$

and $|\mathsf{D}|$. The elements of these are

$$F_{i\overline{j}} = \begin{cases} |Q_i| & \text{if a flux } |Q_i| \text{ enters node } \overline{j} \text{ along edge } i, \\ -|Q_i| & \text{if a flux } |Q_i| \text{ leaves node } \overline{j} \text{ along edge } i, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$|D|_{i\overline{j}} = \begin{cases} 1 & \text{if edge } i \text{ connects to node } \overline{j}, \\ 0 & \text{otherwise.} \end{cases}$$

Combining, we create the matrix $\mathsf{S} = |\mathsf{D}|^t|\mathsf{F}^-|$, where $F^-_{i\overline{j}} = \min(F_{i\overline{j}}, 0)$. The elements of S are

$$S_{\overline{i}\overline{j}} = \begin{cases} \text{flux out of node } \overline{j} \text{ to node } \overline{i} & \overline{i} \neq \overline{j}, \\ \text{total flux out of node } \overline{j} & \overline{i} = \overline{j}. \end{cases}$$

From this we finally obtain the transition matrix

$$P_{\overline{i}\overline{j}} = \begin{cases} S_{\overline{i}\overline{j}}/S_{\overline{i}\overline{i}} - I_{\overline{i}\overline{j}} & S_{\overline{i}\overline{i}} \neq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{22}$$

where I is the identity matrix.

In §4.7.2, it will also be useful to have the edge-to-edge transition matrix. We denote this matrix P^{edge}; its entries P^{edge}_{ij} are the probability a fluid parcel will travel from edge j to i passing through a single node. We calculate this using the matrix approach as follows. Let

$$\Phi_{i\overline{j}} = \begin{cases} 1 & F_{i\overline{j}} > 0, \\ 0 & \text{otherwise}. \end{cases}$$

Each row i of the matrix Φ has entry unity for the node $\overline{j}$ that fluid flowing through edge i enters and zero otherwise. Now let $\mathsf{T} = |\mathsf{F}^-|\Phi^t$; T_{ij} is the flux along edge i away from the node that material in edge j ends up at. Normalizing, we get the edge-to-edge transition matrix:

$$P^{\text{edge}}_{ij} = T_{ij} / \sum_{k=1}^{MN+(M-1)(N-1)} T_{kj}. \tag{23}$$

4.4 Network characteristics

Using the calculated flow field and the transition matrix, the permeability and tortuosity can be found. The permeability k of the network is

$$k = \frac{\text{total flux through network/network width}}{\text{applied pressure difference/network thickness}} = -\frac{q_{\overline{MN+1}} N}{M p_0}.$$

For the network shown in figure 6a $k = 3.39$. The tortuosity τ is defined as the ratio of the expected path length of a fluid particle through the network to the thickness of the network N or

$$\tau = \langle \text{path length} \rangle / N.$$

We use a Markov chain approach to calculate this. Let $\mathbf{x}^{(n)}$ be the vector of probabilities that a fluid parcel is at each node after following exactly n edges. The parcel starts at the top of the network, $\mathbf{x}^{(0)} = (0, 0, \ldots, 0, 1)^t$. After following n edges, the probabilities of being at each node are $\mathbf{x}^{(n)} = \mathsf{P}^n \mathbf{x}^{(0)}$. The probability of following exactly n edges to leave the network is then $\sum_{\overline{j}:\text{sink nodes}} x^{(n)}_{\overline{j}}$. Thus the expected length of a path is

$$\langle \text{path length} \rangle = \sum_{n=0}^{\infty} n \sum_{\overline{j}:\text{sink nodes}} x^{(n)}_{\overline{j}} = \sum_{\overline{j}:\text{sink nodes}} \sum_{n=0}^{\infty} n x^{(n)}_{\overline{j}} = \sum_{\overline{j}:\text{sink nodes}} \left(\mathsf{P} \left[(\mathsf{I} - \mathsf{P})^{-1} \right]^2 \mathbf{x}^{(0)} \right)_{\overline{j}}.$$

For the network shown in figure 6a $\tau = 1.33$.

4.5 Porosimetry

We now turn to simulating the various experiments. In the porosimetry experiment (described in §1) the ultimate goal is to determine the pore-size distribution from measured ΔV vs. p_0 data, where ΔV is the incremental increase in liquid volume drained from the whole network.

The goal of our simulations is to: +

1. Randomly assign d_i values to the edges of a 2D lattice from $\Gamma(1, 2; d)$[†];
2. Compute the ΔV vs. p_0 data that would arise were a porosimetry experiment to be conducted on the 2D lattice;
3. Develop an algorithm to infer the underlying diameter frequencies in the lattice from the computed ΔV vs. p_0 data.

4.5.1 Current model

To simulate a porosimetry experiment, we use the lattice shown in figure 2, with periodic boundary conditions.[‡] Thus, if one were to view an animation of vacated edges[§] a branching chain of such edges that appears to leave the right-hand boundary would simultaneously appear to be re-entering the left-hand boundary.

In an actual porosimetry experiment, the porous medium to be tested is underlain by another porous layer of much lower permeability. The fluid in these tiny-diameter edges at the base of the model never evacuates. To mimic this more accurately, we here treat the "sink" edges in figure 2 as low-porosity substrate edges, *not* as part of the porous medium. (This is accomplished, numerically, by assigning "sink-edge" diameter values that are 1/10 of the minimum diameter found among the regular edges.)[¶]

Note that for the porosimetry simulations M and N are inverted (M is the thickness and N is the width of the sample).

4.5.2 Computing the ΔV vs. p_0 data

Conceptually, the computation which associates increased applied pressure, p_0, with a differential extruded volume, ΔV, is quite straightforward.

To begin, each individual edge is assigned its diameter d_i from a random sampling of the probability-density function $\Gamma(1, 2; d)$. This establishes the critical capillary pressure for each edge, $p_i^c = 1/d_i$. The pressure step-size ($\mathrm{d}p_0$) is set to the smallest difference found among the edges' capillary pressures.

Note that at the start of the simulation the pressure at all nodes is initially zero. Thus, all nodes are initially assigned to the list of nodes having zero pressure save the source node, which is assigned to the list of nodes at the applied pressure p_0.

To compute the simulated porosimetry data, ΔV vs. p_0, the pressure at the source node is incremented in steps $\mathrm{d}p_0$. At each "*istep*"th pressure step the algorithm:

- Determines the pressure drop across each edge;
- Determines which edges' pressure drops exceeds their respective individual critical capillary pressures;
- For each of the edges found to have pressure drops exceeding their critical capillary pressures;
  - computes the corresponding vacated volume and sums this incremental vacated volume into the total differential volume for this pressure step:

$$(\Delta V)_{istep} = \sum_{i\text{: drained edges}} d_i^2$$

  - moves the nodes associated with the newly-vacated links from the list of zero-pressure nodes and to the list of nodes at the applied pressure p_0;
- Sets the pressure of all nodes associated with evacuated edges to the current applied pressure p_0.

[†]The numerical model has, built into it, the capability for randomly sampling *any* input frequency distribution; see the matlab routines CREATE_CUM_DIST() and GETEDGEDIAMETERS(), for example.

[‡]All of the porosimetry-simulation routine can be run with periodic-boundary conditions either enabled or disabled.

[§]The routine POROSIMETRY_MOVIE(20,40,1) provides a 20x40 lattice animation with periodic boundary conditions. Vacated intra-lattice edges are animated as dotted lines. See also figure 8.

[¶]In the porosimetry animation routine, POROSIMETRY_MOVIE(...), the low-porosity substrate edges are visualized with cross-hatches to distinguish them from the regular edges of the porous material. See figure 8.

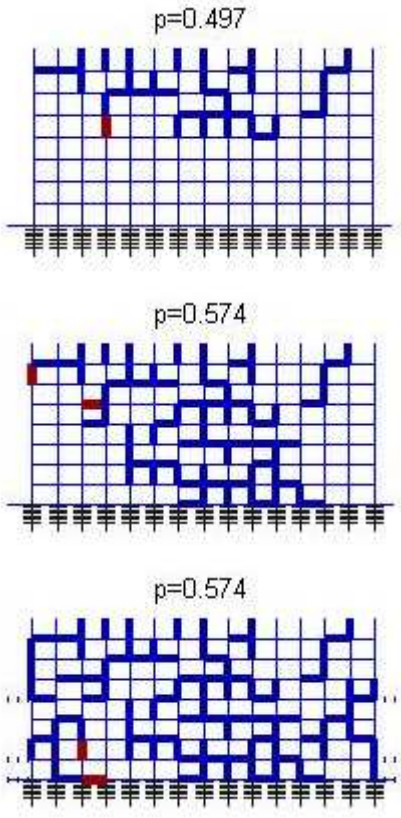

FIGURE 8: *Evolution of vacated edges with periodic boundary conditions and a low-porosity substrate. Blue edges are already vacated; red edges are newly vacating.*

Iteration *must* be carried out in each pressure step. This is because the chain of edges that may vacate at a given pressure is of an unknown length, and only one edge in this chain of vacated pores is "popped" in each iteration. Figure 8 shows a typical evolution of the vacated edges – it has a rather random-walk quality.

4.5.3 Extrapolating the diameter distribution from the ΔV vs. p_0 data

The above-described computation[‖] simulates (in two dimensions) the data that might arise in a porosimetry experiment, ΔV_{istep} vs. $(p_0)_{istep}$.

The goal of this simulation is to gain insight for how best to extrapolate the underlying pore-size distribution from the porosimetry data. Because the precise underlying distribution is known, and used to generate our numerical simulation of the porosimetry phenomena, it provides an excellent opportunity to evaluate strategies for estimating the underlying pore distribution.

Two candidate methods for estimating the underlying pore-diameter distribution from the porosimetry data are now described.

(1) The basic approach

In the basic approach it is assumed that all of the newly-vacated edges in step *istep*, producing a total extruded differential volume ΔV_{istep}, have identical diameters. This identical diameter's value is estimated

[‖]The matlab functions POROSIMETRY_MOVIE(...) and POROSIMETRY_P_VS_VOLUME(...) both use the same computational strategy. However, the first function is produces/saves animations, and the second function instead generates the $[P, dV, V]$ data stream. The POROSIMETRY_P_VS_VOLUME() is called by the main calling routine, PORE_DISTRIBUTION_FROM_POROSIMETRY().

by $d_{istep} = 1/(p_0)_{istep}$. Dividing the differential extruded volume by the volume of the presumed identical edges provides an estimate for lattice frequency of this diameter, $N(d_{istep}) = \Delta V_{istep}/d_{istep}^2$.

It is unsurprising that this approach gives poor agreement between the computed edge-diameter frequencies and the actual underlying distribution. In both the actual porosimetry experiment and in its numerical simulation described above, each subsequent pressure increase to $(p_0)_{istep}$ may evacuate pores with diameters as small as $d_{istep} = 1/(p_0)_{istep}$, but also larger-diameter pores as well. Presuming that all newly-vacated pores in a given pressure step have the same diameter is a weak assumption.

(2) Using the known underlying pore-distribution PDF

In this section we shall illustrate how to recover the underlying pore-diameter distribution from the porosimetry data using both the ΔV_{istep} vs. $(p_0)_{istep}$ data and an underlying probability distribution function (PDF).

Now, one might consider utilizing, say, the known underlying PDF to analyze simulated porosimetry data as a rather pointless activity. In real experiments the PDF is both unknown and precisely the function you are looking for, right? However, this exercise is of value for two reasons.

First, it suggests the "best we can hope for" from the porosimetry experiment. We shall see that, even in this perfect contrived situation, i.e. having: (1) exact numerical data, and (2) perfect knowledge of the underlying PDF, one can accurately recover the frequencies of only large pores (with diameters greater than the mean); the frequency of small pores is vastly underestimated. This, upon reflection, is an expected situation, because smaller-diameter pores which are flanked by larger pores will not necessarily drain, either in the experiment or in the simulation.

Second, this discussion outlines the methodology for convolving porosimetry data with a presumed underlying PDF to compute pore-size distribution frequencies. It emerges that the PDF used to analyze the data need not be the exact underlying distribution to give good results. For example, the use of a generic PDF ($1/d^3$ was tried, motivated by Buckingham-Pi arguments) gives nearly as close a fit as the exact underlying PDF (see figure 9). Consequently, the method is general and may lead to a possibly improved strategy for analyzing porosimetry data.

Finally, it is worthwhile to note that this method to analyze porosimetry data, shown below, is demonstrated on dimensionless simulated numerical data. To carry out this analysis on real porosimetry data, the experimental values would, first, need to be first be scaled using a mean pore diameter $\hat{d}$ which, at this stage, is unknown. Consequently, while the method shown below, on the simulated data, generates a pore-size frequency-distribution in one pass, the corresponding analysis on physical data would necessarily be iterative, with successive improved guesses for the scaling parameter $\hat{d}$ in each iteration.

What follows is a brief outline of the method for estimating the pore-size frequencies from porosimetry data, utilizing a presumptive PDF to distribute the evacuated volume at each pressure step over a range of diameters. We shall accumulate, from our ΔV_{istep} vs. $(p_0)_{istep}$ data, the frequency of diameters in a set of $Nbin$ diameter "bins" setting the number of bins by

$$Nbin = \min(M, \sqrt{M \cdot N}, \text{length}(p_0)), \tag{24}$$

where M here is the number of rows in the lattice, and N the number of columns. Second, since a scaled diameter of value $d = 10$ is an extremus value in most dimensionless distributions, we choose 10 as the maximum dimensionless diameter of interest, and define a fixed set of diameter values as δ_k, where:

$$\Delta\delta = 10/Nbin, \qquad \delta_k \equiv k\Delta\delta, \quad 1 \le k \le Nbin. \tag{25}$$

We accumulate the occurrences of the scaled diameters δ_k in vector of bins, NN_k.

To estimate the lattice occurrences of δ_k from the ΔV vs. p_0 data, the following steps are taken for each data pair $istep:$ +

1. Determine which δ_k is closest to $1/(p_0)_{istep}$. This is our *minimum* diameter size.
2. Assume $\Delta V_{istep} = \mathcal{N} \cdot \left(\sum_{i=k}^{Nbin} (\delta_i)^2 \cdot \Gamma(1, 2; \delta_i) \right)$ and solve for the factor $\mathcal{N}$.

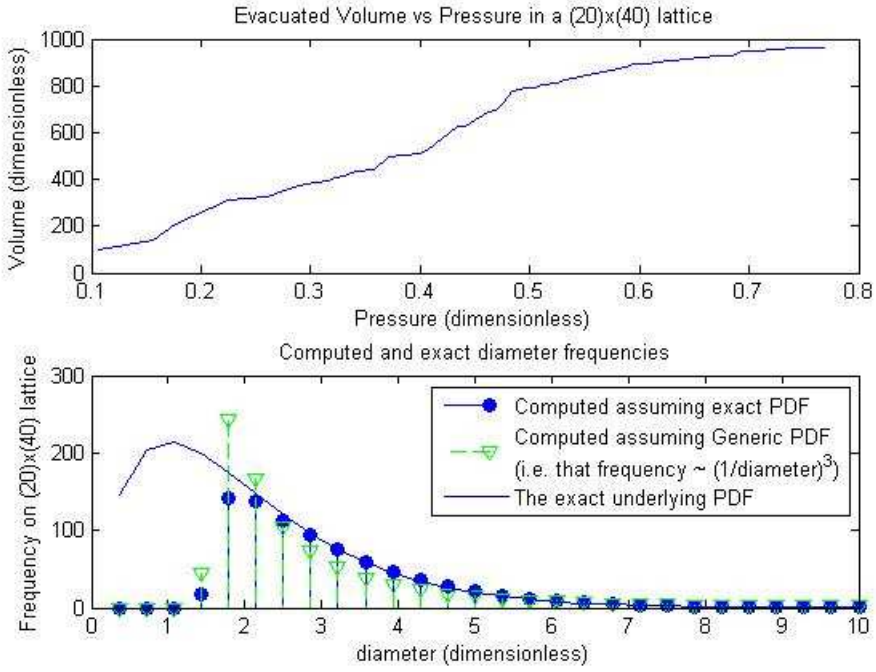

FIGURE 9: *(a) Computed p_0 vs. V data. (b) Comparison between the actual underlying $\Gamma(1,2;d)$ distribution and computed diameter frequencies using (blue dots) the exact underlying PDF and (green triangles) a generic $1/d^3$-tailed underlying PDF.*

3. Update the accumulations for each diameter $i \in [k, ..., Nbin]$ by

$$(NN)_i = (NN)_i + \mathcal{N} \cdot \Gamma(1,2;\delta_i)$$

The *computed* accumulations (i.e., δ_k vs. $(NN)_k$) can be compared to the *exact* underlying distribution (*i.e.*, δ_k vs. $Nedges \cdot \Gamma(1,2;\delta_k) \cdot \Delta\delta$) as shown in figure 9 below. As described above, agreement with the underlying distribution is excellent for the large diameters, but greatly understates the actual distribution of small pores, even though we have used the exact underlying PDF in the evaluation of the data. This is because larger-size pores on either side of smaller pores are vacated earlier in the experiment (at lower pressures) and, regardless of the magnitude of the subsequent applied pressure, there will be no pressure gradient across the smaller pore with which to expel its fluid content.

4.5.4 Porosimetry at small applied pressures

At applied pressures only marginally above the smallest capillary-pressure only a few pores will drain, and we can extend the single-layer results of §3.1 to predict the expected drained volume V at given applied pressure p_0 directly from the pore-diameter distribution.

The probability that a randomly chosen pore will drain when exposed to an applied pressure difference p_0 is again the probability it has a diameter larger than $1/p_0$ or $1 - F_D(1/p_0)$, where $F_D(1/p_0) = \int_0^{1/p_0} f_D(d)\,\mathrm{d}d$ with $f_D(d)$ the pore-size distribution. The expected number of pores in the top layer that drain at applied pressure p_0 is then $M(1 - F_D)$. For each of these open pores, the expected number of "second-generation" open pores radiating away from it is $3(1 - F_D)$ (ignoring any interactions between neighboring pores). For each second-generation open pores, the expected number of third-generation open pores is again $3(1 - F_D)$ and so on. Thus the expected total number of open pores is

$$M(1-F_D) + 3M(1-F_D)^2 + 9M(1-F_D)^3 + \ldots = M(1-F_D)/(1-3(1-F_D)). \tag{26}$$

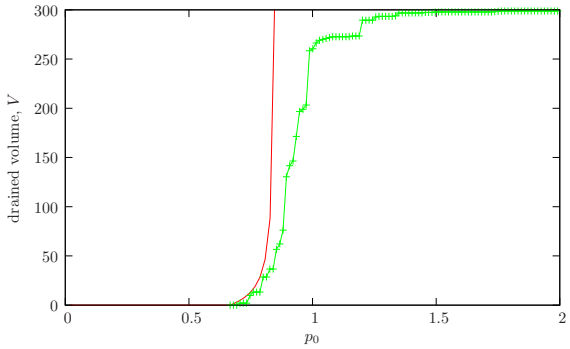

FIGURE 10: *Expected drained volume against applied pressure p_0 (red curve) for a multi-layer porous medium with uniform pore-diameter distribution between* 0.5 *and* 1.5, $M = 20$. *Points are for the network realization shown in figure 6a.*

This prediction is only accurate when a few pores are open, and so the pressure is only marginally above the smallest capillary-pressure, because in reality neighboring pores will interact. At larger applied pressures, (26) provides an upper bound on the expected number of open pores.

We obtain an upper bound on the total drained volume by multiplying (26) by (5). Figure 10 shows the expected drained volume as a function of the applied pressure for a uniform pore-diameter distribution.

4.6 Porometry

The second characterization experiment we consider is porometry (described in §1).

The first measurement to obtain is the bubble point, the applied pressure at which a path first opens through the medium, allowing gas to flow from top to bottom. To find the bubble point in our simulations, we use the identical approach to that of porosimetry (omitting the narrow-pore substrate), and find the pressure p_0 at which a path through the lattice first appears.

The second desired measurement is the gas flux through the system $-q_{\overline{MN+1}}$ at increasing applied pressures p_0. Starting at the bubble point, we increment p_0 by a small amount and calculate the gas flow and pressure distribution for the drained network from the previous pressure. Undrained edges whose capillary pressure is exceeded by the pressure drop calculated are then drained and the calculation is repeated for the updated network until no additional edges are evacuated. The flux through the system is now that required for the given applied pressure.

We calculate the gas flow and pressure distribution in the partially drained lattice using essentially the same approach as for the dry network. To prevent flow through liquid-filled edges, we first (temporarily) set $d_i = 0$ for such edges and then follow §4.1 almost identically. The only difference is that the pressure distribution is now no longer unique at liquid-filled nodes (nodes with no drained edges connecting to them). For this reason, we use the built-in matlab routine `pinv` for the relevant inversions in (19), which gives the correct unique solution at gas-filled nodes and the least-squares solution at liquid-filled nodes (although the latter is irrelevant). The pressure at all nodes is then obtained by setting $p_{\overline{j}} = 0$ at liquid-filled nodes.

A sample total volume flux versus p_0 curve is shown in figure 11. Corresponding images of the drained network and volume fluxes through the edges at various applied pressures are given in figure 12.

4.7 Filtration

Finally, we turn to particle filtration. The measure of filter efficiency that we focus on is retention of particles. Using a stochastic, Monte Carlo-type numerical method, we simulate filter retention dynamics for two different experiments, and complement the second experiment with a Markov-chain description.

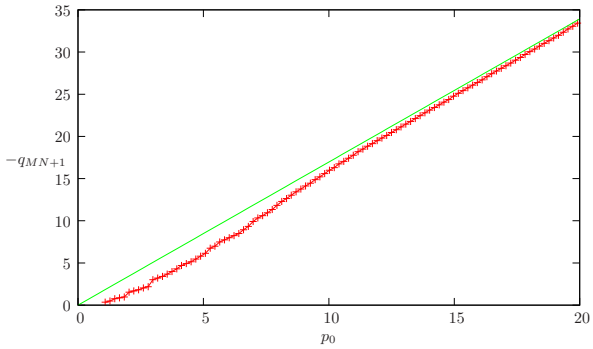

FIGURE 11: *The total volume flux through the network $-q_{\overline{MN+1}}$ against applied pressure p_0 for the network shown in figure 6a. The line is the volume flux through the dry network and the symbols are the porometry measurements.*

4.7.1 Filtration efficiency over time

In the first experiment, we pass particles of a given size through a multi-layered filter. Particle position evolves probabilistically based on the total flux through the filter and the number of pores blocked. Particles that enter a pore of smaller diameter are captured (or retained); hence, as time progresses, pores get increasingly blocked and the filter retention reduces. An algorithm for the numerical procedure is: +

1. Initialize a rectangular lattice network using an incidence matrix connecting nodes to edges (pores) (see §4.1).
2. Compute the probability of particle flow through each pore as a function of the applied pressure drop, pore size, and pores blocked (see §4.3).
3. Select a pore in the top layer of the filter through which the particle will pass, based on the probability of flow through each of these pores.
4. If the particle is smaller than the selected pore, the particle is captured and the pore it blocked. If it is larger, the particle passes through and arrives at the next node, where it selects a pore connected to that node, based on the probability of flow through each of them.
5. This procedure continues until the particle either has been trapped or escapes the filter.
6. The retention is calculated as the fraction of particles trapped over the total number of particles fed. This calculation is done after each particle is either trapped or escapes the filter.

Figure 13 shows the evolution of the filter as particles pass through. As the filter gets increasingly clogged (clogs are represented by red edges), the filter retention reduces. This is shown in figure 14. Note that the initial increase in retention is a result of stochastic effects early in the simulation.

4.7.2 Probability of capture for varying particle size

In the second experiment, we calculate the probability that a particle is captured by a clean filter as a function of particle size. The algorithm for the simulation is the same as for filter retention over time except we clean the filter after each particle passes (*i.e.*, we feed thousands of particles of a single size through a clean filter and calculate the fraction of particles retained). Once sufficient data is collected on a particle size, we change the particle size and repeat the procedure. Figure 15 shows a plot of probability of retention vs. particle size. As one would expect, the retention is good for large particles and gets increasingly worse for small particles.

Alternatively, this efficiency can be calculated using a Markov-chain approach, in a similar manner to the one used for the single-layer theory (see §3.2). As with the single layer theory, our model assumes that particles enter the filter one at a time and do not interact with each other. However, here we assume particles travel through the filter with the flow derived in §4.3.

Consider a particle of diameter d_p. The particle is introduced into one of the top pores connected to the

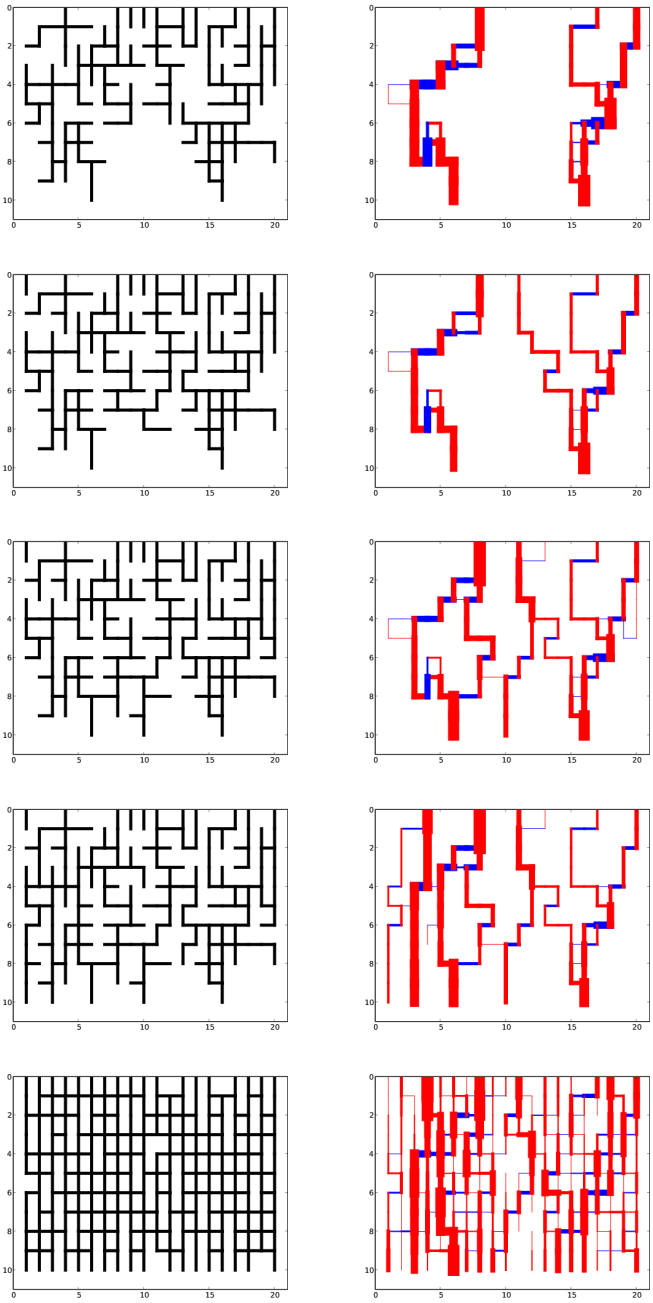

FIGURE 12: *Progression of drained pores (left) and volume fluxes (right) with increasing applied pressure (a)* $p_0 = 1.08$*, (b)* 1.27*, (c)* 1.85*, (d)* 2.80*, and (e)* 19.9*. The network realization is that shown in figure 6a. In the right panels, line width reflects the relative magnitudes of the fluxes through the edges. Red indicates flow with the assigned direction, blue is against. Some pores have such small flux that the discretized line width is zero.*

source node. The particle chooses which pore to enter next based on the probabilities in the edge-to-edge transition matrix (23) and continues choosing subsequent pores and moving through the filter until it is

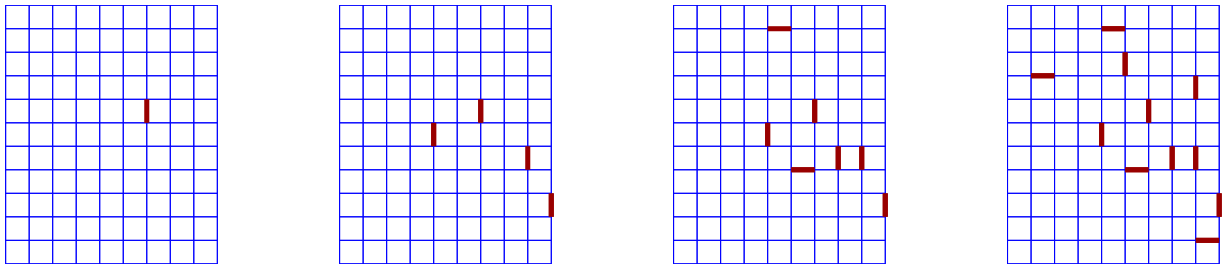

FIGURE 13: *Filter evolution as particles pass through and clog pores. Logarithms of pore-diameters are normally distributed with mean zero and variance one. Particle has diameter* 0.8. *Red edges indicate blockages.*

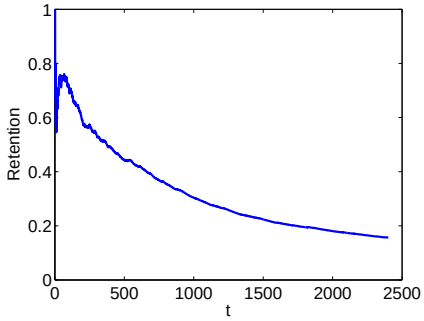

FIGURE 14: *Evolution of filter retention (ratio of captured particles to total number of particles fed to the filter) against "time" (total number of particles fed). Logarithms of pore-diameters are normally distributed with mean zero and variance one. Particle has diameter* 0.8.

either captured by a pore with smaller diameter or it exits the filter through the sink nodes. We calculate the probability the particle is captured for a given realization of the filter.

Denote the set of sink pores (connecting to the sink nodes) by C_1, the set of capturing pores by C_2 (the pores having diameter less than d_p), and the remaining transient pores, including the starting pores at the top of the filter, by T. For the remainder of our calculation to work, C_1, C_2 and T, must be disjoint. That

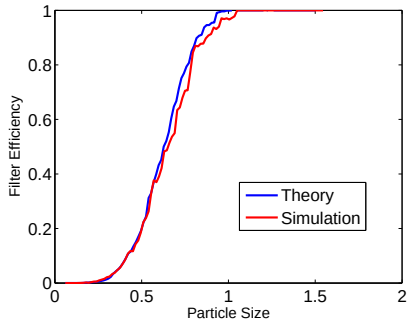

FIGURE 15: *Fraction of particles caught when particles of varying sizes are passed through a clean filter. Logarithms of pore-diameters are normally distributed with mean zero and variance one.*

is, starting pores and sink pores cannot also be capturing pores. Ideally one would like to add another single "pore" at the top and bottom simply for the purpose of calculation, but in a filter with many layers, the top and bottom rows do not significantly affect the result, as will be seen.

To calculated the probability of passing through the filter and the probability of being captured, the edge-to-edge transition matrix, P^{edge}, is shuffled into the form

$$\begin{pmatrix} \mathsf{A}_1 & 0 & \mathsf{S}_1 \\ 0 & \mathsf{A}_2 & \mathsf{S}_2 \\ 0 & 0 & \mathsf{Q} \end{pmatrix}. \tag{27}$$

Here all entries from P^{edge} corresponding to pores in C_1 appear in the matrix A_1 and those corresponding to pores in C_2 appear in matrix A_2. Those entries corresponding to the transient pores T appear in matrices S_1, S_2 and Q, where the ending pores are in C_1, C_2 and T respectively. Zeros in (27) denote block matrices of zeros.

The probabilities of ending up at each pore of C_1 and C_2 are given by [1, §3.3]

$$\mathsf{U}_1 = \mathsf{S}_1(\mathsf{I} - \mathsf{Q})^{-1} \tag{28}$$

and

$$\mathsf{U}_2 = \mathsf{S}_2(\mathsf{I} - \mathsf{Q})^{-1}, \tag{29}$$

respectively. Here the element $(U_k)_{ij}$ is the probability of ending up at the ith pore of class C_k starting from the jth pore of class T.

The particle starts in one of the pores in the top row, thus the initial vector of probabilities of being at each pore of T

$$\boldsymbol{\phi}_0 = [0,\ \ldots,\ 0,\ f_{\overline{1}},\ f_{\overline{2}},\ \ldots,\ f_{\overline{M}}]^t, \tag{30}$$

where $f_{\overline{j}}$ is the probability to pass from the top node $\overline{MN+1}$ to node $\overline{j}$ (element $P_{\overline{jMN+1}}$ of (22)), equivalently the probability the particle starts in the top pore connecting to node $\overline{j}$. The probability of passing through the filter and the probability of being captured are then the weighted sums of U_1 and U_2:

$$P(\text{exit}) = \sum_{i,j} (U_1)_{ij}(\phi_0)_j, \qquad P(\text{captured}) = \sum_{i,j} (U_2)_{ij}(\phi_0)_j, \tag{31}$$

respectively. The filter efficiency for a particle of size d_p is $P(\text{captured})$.

We compare the Markov chain results with simulations in figure 15. Plots are obtained by repeating the above analysis for many different particle sizes, but the same realization of the filter. The slight discrepancy is due to eliminating top pores and bottom pores as capturing pores in the analysis.

Note that we apply this model to a given realization of the filter where all pore sizes have been selected from the appropriate diameter distribution, $f_D(d)$. Our analysis does not average over different model filter realizations, but this average is effectively done by considering a filter large enough.

4.7.3 Sparse capturing pores

The simple single-layer model (*cf.*, §3) has important implications to multi-layer filters where capturing pores are rare. The key attribute of a single-layer filter is that its capturing pores do not interact, *i.e.*, flow through a given capturing pore can not pass through a second capturing pore. Thus the probability that the next particle is captured by the filter, P_{capture}, is given by a simple flow ratio:

$$P_{\text{capture}} = \frac{\text{Flow through Capturing Pores}}{\text{Total Flow through Filter}} \tag{32}$$

Figure 16 illustrates that this attribute of single-layer filters also holds for some multi-layer filters. Suppose for a given filter we can find a cut-line crossing all its capturing pores such that for every pore intersecting

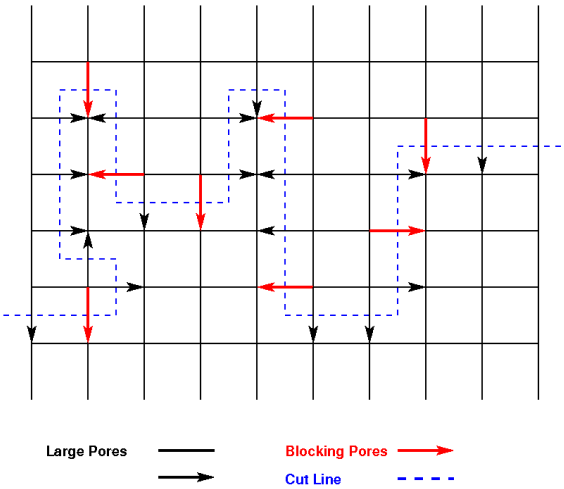

FIGURE 16: *Illustration of a filter with capturing pores which effectively line up in a single layer (defined by the blue cut line). For a given particle, black pores are large enough for the particle to pass through; red pores are small (narrow) and can capture the particle. All the flow crosses the single-layer cut line in one direction. The direction of flow in the large pores that do not intersect the cut-line does not affect* P_{capture}.

the cut-line, the flow crosses the cut-line in the same direction. Then the entire flow crosses the effectively single layer of the cut-line, and (32) still holds for the filter.

In many actual filters, sparse capturing pores may not be completely non-interacting; flow paths could conceivably pass through two or more capturing pores in which case the required cut-line could not be found. But since the capturing pores are sparse in the filter, the probability of these pores interacting may be extremely low, making a single-layer approximation possible. Thus in the sparse-capturing-pore case, the capture probability is still given approximatively by (32).

5 Summary

We have created a framework and numerical code for simulating dry flow, porosimetry, porometry, and filtration through an idealized filter consisting of a 2D rectangular lattice of cylindrical capillaries. The results of these simulations could be used as input to analysis techniques used in experiments for inferring the pore-size distribution. Comparisons of underlying and calculated pore-size distributions would then illuminate the reliability of these experimental techniques. We have already shown that porosimetry underestimates the number of small pores, even when our data is exact.

For a single layer of pores we could predict the expected observations directly from the underlying pore-diameter distributions. Thus for a sufficiently large, thin layer we can invert the experimental measurements to obtain a pore-size distribution directly.

A particular shortcoming of our porosimetry and porometry simulations is that they are static: if a set of several interconnected pores could open once a bottle-neck pore eventually drains, then the dynamics of how these pores open is important in determining which of the set do drain. Such dynamic evolution was beyond what was considered at the workshop but should perhaps be explored.

References

[1] S. KARLIN AND H. M. TAYLOR, *A first course in stochastic processes*, Boston, second ed., 1975.