

MTM (Mass Transfer Matrix) Espresso extraction model

Andrew Zaporozhets, April 2020

This document describes an espresso extraction model which is influenced by the work presented in [1] and [2]. The model uses the coffee flow data collected during the espresso shot and the grind parameters along with other assumptions to predict the TDS (total dissolved solids) in the coffee drink. The model needs to be calibrated first to the TDS measurement data.

The main differences of this model compared to [2] is that a) this model uses directly the coffee flow data (which can be collected with e.g. Bluetooth scale) instead of trying to simulate it. This improves the accuracy of the model. And b) that the model uses natural discretisation of the coffee particles into cells to model the extraction dynamics, instead of using partial differential equations which are then discretised for numerical solution. With this approach the particle size distribution can be used directly to create families of particles with different sizes. This improves the accuracy of the model further.

The model is implemented in C# <https://github.com/AndyZap/EspMod> and is designed to work with DE1 espresso machine shot files.

References

1. K. M. Moroney, Heat and mass transfer in dispersed two-phase flows, Ph.D. thesis, University of Limerick (2016). <https://ulir.ul.ie/handle/10344/6592>
2. M. Cameron, D. Morisco, D. Hofstetter, E. Uman, J. Wilkinson, Z. Kennedy, S. Fontenot, W. Lee, C. Hendon, J. Foster, Systematically Improving Espresso: Insights from Mathematical Modelling and Experiment. Matter 2, 1–18 March 4, 2020, <https://doi.org/10.1016/j.matt.2019.12.019>

Possible use of the model

The coffee TDS measurements are common in assisting the search to a good brew recipe. The modern coffee brewing methods involves considerable range of the brew ratios, pressure and flow profiles. The TDS values obtained by different methods might be difficult to relate to each other. An espresso extraction model could tell if the differences are the first order effects, predictable by the model, or something more complicated (and possible undesirable) is going on – e.g. like channelling.

Model description

Following [1] and [2], we consider water flow through a cylindrical packed coffee bed (the puck).

A grinder breaks coffee beans into particles. For an espresso grind, a typical particle size is about 200-300 μm . An interesting feature of coffee particles is that they are composed from a little hollow cells with size around 20-50 μm . [1] estimates that the void volume inside cells is about 64% of the

total particle volume. In addition to the larger particles (the boulders), an espresso grind contains “fines” – small particles about the size of a coffee cell. These are usually the broken cell fragments.

A good grinder usually produces about 10-15% of fines from the total grind mass/volume. Also the particle size distribution of the boulders is rather narrow, i.e. dominated by particles of a particular size. Check [1] and [2] for typical particle size distributions graphs.

Figure 1 shows a schematic representation of a coffee bed. For simplicity, we represent coffee particles as spheres. Also we assume that all little coffee cells are of the same size and packed into concentric rows with the width equal to the cell size. Note the cells are packed tightly without any gaps, e.g. imaging cells in a pomegranate fruit. Following these assumptions, the particles can only be composed of a certain number of concentric cell layers, and the particle diameter can only take a set of discrete values: 1 cell size for 1 cell layer (i.e. 1 cell), 3 cell sizes for 2 cell layers, 5 cell sizes for 3 cell layers, etc (see the figure).

The space between particles is easily accessible by water. This is the path the water takes during the espresso extraction. Reference [1] suggests that the particles occupy about 83% of the puck volume, so the remaining 17% is the space between particles.

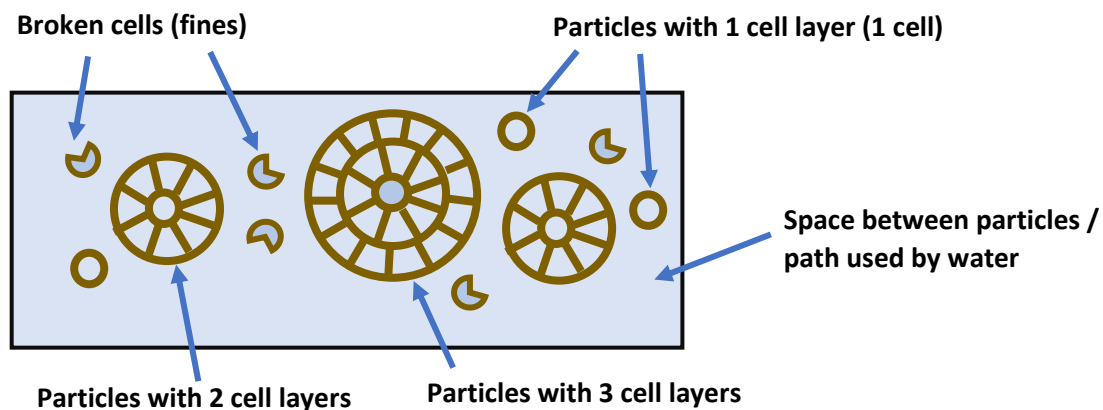


Figure 1. Schematic representation of a coffee bed as spherical particles of different sizes.

At the beginning of the espresso extraction the puck is filled with water (the pre-infusion stage). The water fills not only the space between particles, but also the space inside the coffee cells due to the numerous micro-cracks created during the roasting process. So from the total puck volume we have about 17% water volume between the particles and $83\% \times 64\% = 53\%$ inside the particle cells. The remaining volume is occupied by the coffee solids. Just to remind that we assume that there is no space between cells inside a particle.

Next let us turn to the part of the coffee particles which is soluble by water. It is estimated that about 30% of the ground coffee mass is soluble. What interesting, is that the soluble part dissolves in water rather quickly - in a few seconds. This means that the soluble mass in the broken cells / fines quickly dissolves into the water volume between particles.

It needs to be noted that the soluble part actually contains around 2,000 different compounds extracted from coffee grounds during brewing. But following [1] and [2], we model the total soluble coffee mass dynamics as some average quantity. The model could be possibly rewritten to work with individual compounds, but such model would be difficult to calibrate to experimental data, as it

difficult to measure the amount of an individual compound in the coffee brew. From the other hand, the total dissolved solids (TDS) can be easily measured with a coffee refractometer, so a model formulated for the total soluble coffee mass can be calibrated without any difficulties.

Diffusion through the cell boundaries

The soluble coffee part inside cells dissolves quickly as well. But it is far more difficult for the coffee mass to migrate into the space between particles where it could be picked by the moving water. This is a diffusion process and can be described using Fick's first law of diffusion:

https://en.wikipedia.org/wiki/Fick%27s_laws_of_diffusion#Fick's_first_law. The law says that the speed of the mass transfer across the cell boundary (e.g. in gram per second) is proportional to the difference in the coffee mass concentration (e.g. in gram per mm³) on both sides of the boundary.

For the mass transfer between two cells the formula can be written as:

$$dm = K^* \cdot (c_1 - c_2) \cdot dt = K^* \cdot \left(\frac{m_1}{v_{cell}} - \frac{m_2}{v_{cell}} \right) \cdot dt \quad (1)$$

Here dm is the amount of mass transferred over the cell boundary during the time interval dt , m_1 and m_2 , c_1 and c_2 is the soluble coffee mass and the coffee concentration inside cell 1 and cell 2 correspondingly. v_{cell} is the void volume inside coffee cell which contains water and soluble coffee mass. K^* is a coefficient which describes the rate of diffusion between two cells. It is proportional to the diffusion coefficient and the area of the cell boundary, as they appear in the Fick's formula. As we assume that all cells are of the same size, we do not introduce a variable for the area of the cell boundary, and simply include it implicitly in the value of K^* . Here we use v_{cell} with small letter "v" for the void volume inside coffee cell, and will later use V_{cell} with capital "V" for the total cell volume. As per [1], the void volume inside coffee cell is about 64% of the total cell volume.



Figure 2. Diffusion mass transfer between two coffee cells

Instead of the coefficient K^* it is actually easier to work with another coefficient which is scaled by the void volume inside cell v_{cell} :

$$K^* = K \cdot v_{cell} \quad (2)$$

The equation (1) can now be rewritten as:

$$dm = K \cdot (m_1 - m_2) \cdot dt \quad (3)$$

K is the only coefficient used by the model which needs to be fitted to the experimental TDS measurements.

With $dt=1$ (second) and $K = 0.5$ we can see that the mass transferred over the boundary is equal to the half of the mass difference. I.e. after 1 second time step the mass in cells 1 and 2 become equal, so the diffusion process is complete. From the coffee brewing we know that it takes tens of

seconds to extract coffee, so we might expect K values about an order of magnitude smaller. So the way we setup the equation 3 helps with the interpretation of the K coefficient.

Next let us write the formula for the diffusion between a coffee cell in the outer layer and the space between coffee particles. This is the final destination of the diffusion process, where the soluble mass can be carried to the cup by moving water. We assume that the interface area is about the same size as between two cell, so we use the same coefficient K :

$$dm = K \cdot v_{cell} \cdot (c_1 - c_{between}) \cdot dt = K \cdot v_{cell} \cdot \left(\frac{m_1}{v_{cell}} - \frac{m_{between}}{v_{between}} \right) \cdot dt \quad (4)$$

Here $c_{between}$ and $m_{between}$ is the concentration and the soluble coffee mass in the space between particles, $v_{between}$ is the volume between particles.

Tracking the coffee mass changes in the cells during extraction

Tracking soluble coffee mass in every coffee cell makes the numerical solution very difficult, as the number of cells in a typical espresso puck is very large. Let us make some assumptions to make the model less numerically demanding.

We already assumed that all cells are of the same size and all have the same internal volume. Next let us assume that all cells have the same initial soluble coffee mass, hence the same initial coffee mass concentration.

As with many numerical techniques, let us discretise the solution domain (the puck) into a set of S thin horizontal slices (here we assume that the water moves from the top to the bottom of the puck, as in an espresso machine). The coffee concentration within a slice is assumed to be constant, so it needs just one variable $m_{between, s}$ to track the soluble coffee mass between the particles in a slice number s .

Next let us turn to the spherical particles which are built from concentric cell layers (see **Figure 1**). As the initial coffee concentration within cells is the same for all cells, and the coffee concentration is constant within the puck slice, then it is enough to have one variable per cell layer to track the mass for all cells within this layer. E.g. for all particles with 3 cell layers in a given puck slice we need only 3 variables to track the coffee mass. Note that for all particles with e.g. 4 cell layers we need another 4 variables, as the diffusion dynamics is different for cells with 3 and 4 layers.

So the idea to discretise the puck into slices and consider cell layers within a coffee particle which all follow the same diffusion dynamics reduces the number of variables considerably and makes the numerical implementation not so challenging.

In the software we introduce a matrix with 3 indexes: s (the puck slice number), p (the particle size index) and l (the cell layer number within the particle) – to track the soluble mass within cells. Then we use a set of rules to update the mass values within the matrix. This is why we call this approach the mass transfer matrix espresso extraction model.

Rules for the mass matrix update

We need to make some changes to Equation 3 and 4, to account for the number of cells within each cell layer.

This is how we calculate the number of cells per each layer. We calculate a single cell volume using the formula for a sphere:

$$V_{cell} = \pi \cdot \frac{X_{cell}^3}{6} \quad (5)$$

where X_{cell} is the coffee cell size. We are a bit vague what is the cell size is as we have not defined the cell shape exactly. All we want is that all cells have the same volume and to be packed tightly into concentric layers of depth X_{cell} to form a particle. This is how we intend to use this variable. The reason for using the formula for a sphere to calculate a single cell volume is because we assume that the particles are spheres, so the volume for a particle with 1 cell layer (i.e. with 1 cell) is equal to 1 cell volume. If we used e.g. a formula for the volume of a cube for a single cell, this would not be the case.

As we discussed before, the particle diameter $D_{particle}$ can take a set of fixed values only:

$$D_{particle} = X_{cell} + 2 \cdot X_{cell}(N_{layers} - 1) \quad (6)$$

N_{layers} is the number of layers in a particles. For the particle volume we use the same formula for a sphere:

$$V_{particle} = \pi \cdot \frac{D_{particle}^3}{6} \quad (7)$$

And the number of cells per particle:

$$N_{cells \text{ per particle}} = V_{particle} / V_{cell} \quad (8)$$

To calculate the number of cells per layer let us look at the difference between number of cells for two particles, where the number of layers is different by one. All these extra cells are places in the extra layer, i.e. this difference gives the number of cells per top layer. Working iteratively from a particle with 1 cell we can calculate the number of cells per layer. **Table 1** shows calculations for cell size $X_{cell} = 30 \mu\text{m}$, the value used in [1].

Number of cell layers	Particle diameter, mm	Particle volume, mm ³	Number of cells per particle	Number of cells in the top layer
1	0.03	1.41372E-05	1	1
2	0.09	0.000381704	27	26
3	0.15	0.001767146	125	98
4	0.21	0.004849048	343	218
5	0.27	0.010305995	729	386

Table 1. Calculations of the number of cells per particle and per cell layer.

Now let us consider diffusion between two adjacent layers with number of cells per layer N_{layer1} and N_{layer2} . Assume that layer 2 is further away from the particle center, so has more cells. The equation (3) is written for 2 cells “connected” to each other. In the case we consider here one cell from layer 1

is “connected” to N_{layer2}/N_{layer1} cells on layer 2. So when the mass in a cell in the layer 1 changes by dm , the corresponding change of mass in a cell in the layer 2 is equal to $-dm \cdot N_{layer1}/N_{layer2}$. The outer cell boundary (i.e. the one which is further away from the center of the particle) is the “reference” for the mass calculation, and the inner cell boundary (closer to the center) requires scaling.

The choice which layer has mass change dm and which one has a corresponding change scaled by the ratio of the number of cells is arbitrary. I.e. we could have chosen layer 2 cell mass change equal to dm . This choice results in slightly different values for the coefficient K when fitted to the same experimental TDS data, as we include the “reference” cell interface area in the value of K . This does not impact the overall logic of the algorithm.

Next, the equation (4) correctly gives the mass change in a cell in the outer layer, as we consider the outer cell boundary as the reference. To calculate the change to $m_{between, s}$ (the soluble coffee mass in the space between particles, per puck slice number s) we need to multiple the dm value from the equation (4) by number of particles of a given size per puck slice, and by number of cells in the outer layer for this particle size. This gives the total number of cells “connected” to the space between particles in the puck slice.

Finally, let us consider the water flow, which carries soluble coffee mass between puck slices down to the cup. Assume that in time interval dt the pumps adds fresh water with volume V_{water} to the top slice of the puck. The same volume of liquid is pushed from the top slice to the next slice below. But this liquid contains water and some soluble coffee mass which can be calculated as:

$$dm_s = V_{water} \cdot c_{between,s} = V_{water} \cdot \frac{m_{between,s}}{v_{between,s}} \quad (9)$$

This equation is used for each puck slice s to obtain the mass change due to the water flow. Note that the total mass change per slice is equal to the mass received from the upper slice (0 for the top slice) and the mass pushed to the layer below.

The liquid from the bottom slice is pushed into the cup. So we update the variables to track the total volume in the cup and the soluble coffee mass in the cup as:

$$V_{cup} += V_{water} \quad m_{cup} += dm_s \quad (10)$$

Here we use the commonly used programming notation “+=” for a variable increment.

Numerical algorithm

At the beginning of the algorithm we calculate the required variables to setup the simulation (see the next sub-section for the formulae and examples).

The numerical algorithm works in time steps dt starting with $Time = 0$. We repeat the same set of calculations during each time step, and at the end of the step increase the time to the next step $Time += dt$, until we reach the specified time of the shot.

First we calculate the amount of water pushed into the puck during the time step, i.e. from $Time$ to $Time+dt$. We use the Decent machine shot files as input, and read two arrays: “espresso_elapsed” (time in sec) and “espresso_weight” (weight in the cup, gram). From these arrays we calculate the mass of the liquid added to the cup. The liquid contains water + soluble coffee mass. We do not know the coffee mass added during this time step yet, but can approximate this amount but taking

the value calculated during the previous time step. This is common for numerical algorithms to take the values which are not known yet from the previous step. Assuming that the time step is small (typically, a fraction of a second), and the extraction rates does not change sharply, this provides a good approximation. Subtracting the soluble coffee mass from the total mass added to the cup we arrive at the water mass. This mass is converted into water volume using the density of the water.

Then we proceed with the mass matrix update using the equations (3) and (4) with the corresponding scaling coefficients, as described in the section above. We iterate over 3 indexes: s (the puck slice number), p (the particle size index) and l (the cell layer number within the particle). For example, we can have 10 puck slices, 2 particle sizes (e.g. with diameter 0.21 and 0.27 mm) and the corresponding number of cell layer per particle (4 for 0.21 mm particle and 5 for 0.27 mm particle). Also we calculate the values for the coffee mass between grains per puck slice, and the coffee mass in the cup. The mass changes dm are not applied immediately to a corresponding matrix entry. Instead, we calculate all mass changes for all elements first (this requires a separate storage matrix). Then, at the end of the time step, all mass values are updated using the mass changes values. This ensures the consistency between all values, so the calculation does not depend of the order in which the values are updated.

At the end of the calculations the algorithm report the TDS value in the cup, given the total water volume and the soluble coffee mass in the cup.

The software can run in two modes: a simpler and faster simulation mode - i.e. when the value for the coefficient K is provided in the input. And a slower calibration mode, when the measured value for the TDS is given. In this case the simulation mode is called repeatedly inside a root-finder algorithm, to find the matching value for the coefficient K .

The remaining variables and formulae to setup the simulation

Table 2 lists the variables used and their typical values, as per [1] and [2]. The formulae used to calculate the values are given after the table.

Description	Value	Name in the software	Notation in equations
Basket / puck diameter, mm	58.5	puck_diameter_mm	D_{puck}
Coffee cell size, mm	0.03	coffee_cell_size_mm	X_{cell}
Bean weight, g	18	dsv2_bean_weight	M_{bean}
Horizontal puck cross-section area, mm ²	2688		
Puck volume, mm ³	54546		
Puck height, mm	20.29		

Table 2. List of variables and their typical values, as per [1] and [2].

$$K^* = K \cdot V_{cell} \quad (2)$$

Example simulation results

sdfsdfs

