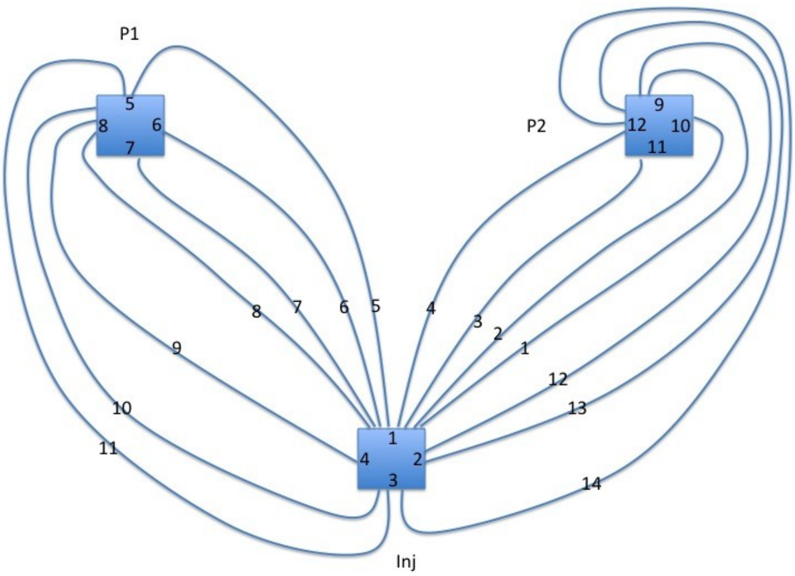
# Optimizing streamline flux assignments (fluxopt)

## *The example*

The configuration we looked at is as in the figure. It has 3 wells (one injector and two producers) in 2D. Shown are 14 streamlines. Note that these are really streamline clusters: each face-to-face connection is represented by only one streamline. In reality there will be more streamlines per cluster.  In this setting, with the chosen face-to-face connections, the maximum number of unique clusters is the given 14.




## *General overview of the approach*

Let's call the total number of well faces nwf, the faces are denoted by f(i), the streamlines by j, j=1, …, nsl, and the streamline fluxes by slf(j).
We assume that each f(i) has a unique total flux F(i) in or out of the face (so, we are not assuming that for each well, the total well rate is equally divided over its faces).
We also assume that each well face has streamlines exiting or entering it (which will be the case in practice as all grid cells are required to be crossed by at least one streamline).

To simplify the calculations we assume that all fluxes, be it well face fluxes or streamline fluxes, are positive. When needed, we negate producer fluxes in the system of equations.

For each well face f(i), denote by Si the set of streamlines j that start or end at this well face. For each face f(i), we set up an equation of the form

$$\sum_{j \in S_i} fsl(j) = F(i)$$

This leads to a matrix-vector system Ax=b, with the vector x containing the slf(j)'s and the vector b the fluxes F(i).

The rank of the matrix A will always be equal to the nwf-1. It is not nwf because any one of the face fluxes is in fact dependent on the others (it is the total reservoir flux – sum of the other nwf-1 well face fluxes). The right hand side b has that dependency in it as well of course.
What this means is that Ax=b will always have an exact solution. However, this solution is not useful in general as it likely returns several negative slf(j).

In order to solve Ax=b optimally, we need to add positivity constraint for the solutions slf(j). This can be done, for example, by lsqnonneg in MATLAB. Using a non-negative least squares method such as this also easily allows additional constraints to be added, or constraints to be weighted nonuniformly.

What's the procedure?

1. Set up Ax=b
2. Perform a QR to find the relevant streamline clusters (this bundles all streamlines that go from the same face to the same face. In 2D, such streamlines will be adjacent to each other on a face. In 3D they will likely be in the same local area, but may not be fully adjacent)
3. Solve the reduced system for the streamline cluster fluxes.
4. Knowing the total flux for each streamline cluster, now divide that flux over the streamlines in each cluster.

Note that in step 4, we have complete freedom. It could be done to ensure that neighboring clusters (eg one on one face close to a corner and the other just around the corner on the next face) transition smoothly. This will, we think, reduce mapping errors a great deal (because then we avoid having sudden changes in volume in streamlines that leave just around the corner from eachother and can end up in similar cells).

## *Adding constraints*

The above method tries to match the face fluxes optimally. The end result does not guarantee, of course, that the total volume fluxes for individual injectors or producers or satisfied only that the well face fluxes are matched optimally.

However, the method allows the addition of constraints. For example, we can add a total flux constraint for each well. We can also add an equation that gives the total flux of all injectors together (which would force total volume balance). Adding these additional constraints will reduce the mismatches for the total well fluxes.

## *Weighting your favorite constraints*

So far, we have not given any more weight to one constraint or another. The methods allows us to do this however.
We can create a diagonal matrix W and instead find the best nonnegative solution to WAx= Wb. Here wii, the ith diagonal element of W, gives the weight imposed on constraint i. For example, if we want to create a solution that preserves total volume

injected and contraint k is the corresponding added constraint, a large wkk relative to the other wii's will force this constraint to be satisfied.

## *Preliminary tests*

We now consider two test cases for the example problem.

**Case 1:**
Here, each face of a well carries a quarter of the total well flux. The total injector flux is 100, the total flux of producer 1 is 40 and producer 2 has a flux of 60.

We compare

1. The current 3DSL procedure of distributing the injector face fluxes equally amongst the streamlines leaving each face
2. The non-negative least squares procedure (lsqnonneg) with equations for each well face and no additional constraints
3. lsqnonneg with the additional constraint for total injector flux
4. lsqnonneg with the additional constraints for total injector and producer fluxes
5. Same as 4 with the additional constraints weighted 100 times stronger than the others.

Fluxes for each streamline are (oops, I don't know where the first numbers in the last column went…)

| streamline | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| 1 | 3.1250 | 0 | 0 | 0 | |
| 2 | 3.1250 | 10 | 10.3714 | 11.0281 | |
| 3 | 3.1250 | 10 | 10.3714 | 11.0281 | |
| 4 | 3.1250 | 0 | 0 | 0 | |
| 5 | 3.1250 | 0 | 0 | 0 | |
| 6 | 3.1250 | 5 | 5.3714 | 4.8007 | |
| 7 | 3.1250 | 5 | 5.3714 | 4.8007 | |
| 8 | 3.1250 | 0 | 0 | 0 | |
| 9 | 25 | 17.5 | 18.4286 | 18.2292 | |
| 10 | 8.333 | 0 | 0 | 0 | 0 |
| 11 | 8.333 | 12 | 12.7429 | 12.0924 | 12.1053 |
| 12 | 12.5 | 17 | 18.3197 | 18.3197 | 18.6842 |
| 13 | 12.5 | 6 | 6.0462 | 6.0462 | 6.0526 |
| 14 | 8.333 | 11 | 12.2736 | 12.2736 | 12.6315 |

The corresponding residuals (assigned – actual)

| Well-face | (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|---|
| 1 | 0 | 5.0000 | 6.4857 | 6.6576 | 7.1052 |
| 2 | 0 | -2.0000 | -0.8857 | -0.6341 | -0.2632 |

| | | | | | |
|---|---|---|---|---|---|
| 3 | 0 | -7.5000 | -6.5714 | -6.7708 | -6.5790 |
| 4 | 1.4583 | 2.0000 | 2.7429 | 2.0924 | 2.1053 |
| 5 | -6.8750 | -5.0000 | -4.6286 | -5.1993 | -5.2631 |
| 6 | -6.8750 | -5.0000 | -4.6286 | -5.1993 | -5.2631 |
| 7 | 26.4583 | 7.5000 | 8.4286 | 8.2292 | 8.4210 |
| 8 | 0.6250 | 2.0000 | 2.7429 | 3.3197 | 3.6842 |
| 9 | -11.8750 | -5.0000 | -4.6286 | -3.9719 | -3.6842 |
| 10 | -11.8750 | -5.0000 | -4.6286 | -3.9719 | -3.6842 |
| 12 | 8.9583 | 2.0000 | 2.7429 | 3.3197 | 3.6842 |
| Inj | 0 | -6.5 | -1.8571 | -1.3813 | -0.0002 |
| P1 | 14.1667 | -0.5 | 1.9143 | -0.0770 | <0.0001 |
| P2 | -14.1667 | -6.0 | -3.7714 | -1.3043 | -0.0002 |

Clearly, we can make the method volume conservative (see results for 5). There are of course mismatches for the faces, but not in the total volume coming in to each producer.
It is possible with weights to, for example, make sure that important faces are well represented.

**Case 2:**
The injector has face data (70,10,10,10) and the two producers are as in case 1. This case is a little more realistic because F(1) is large, and f(1) indeed has most streamlines emanating from it.

In this case the exact solution is in fact non-negative and all the variants of the method yield the same result, listed in the second column below. The first columns is again the 3DSL procedure.

Streamline fluxes are

| streamline | (1) | (2) |
|---|---|---|
| 1 | 8.7500 | 10.0000 |
| 2 | 8.7500 | 15.0000 |
| 3 | 8.7500 | 15.0000 |
| 4 | 8.7500 | 0 |
| 5 | 8.7500 | 10.0000 |
| 6 | 8.7500 | 10.0000 |
| 7 | 8.7500 | 10.0000 |
| 8 | 8.7500 | 0 |
| 9 | 10.0000 | 10.0000 |
| 10 | 3.3333 | 0 |
| 11 | 3.3333 | 0 |
| 12 | 5.0000 | 5.0000 |
| 13 | 5.0000 | 5.0000 |
| 14 | 3.3333 | 10.0000 |

with residuals for each well-face and the residual in the total flux of the injector, (Inj) and the residual in the total flux of the producer (P1,P2) in the following table

| Well-face | (1) | (2) |
|---|---|---|
| 1 | 0 | 0.0000 |
| 2 | 0 | -0.0000 |
| 3 | 0 | -0.0000 |
| 4 | 0 | 0.0000 |
| 5 | 2.0833 | -0.0000 |
| 6 | -1.2500 | 0.0000 |
| 7 | -1.2500 | 0 |
| 8 | 12.0833 | 0.0000 |
| 9 | -1.2500 | 0.0000 |
| 10 | -6.2500 | 0.0000 |
| 11 | -6.2500 | -0.0000 |
| 12 | 2.0833 | 0.0000 |
| Inj | -0.0000 | 0.0000 |
| P1 | 11.6667 | 0.0000 |
| P2 | -11.6667 | 0.0000 |

We get a REALLY NICE SOLUTION!!!!!!!!!

Since this seems to be a more realistic example in which a larger number of streamlines corresponds with a larger face flux we are pretty optimistic that in a real case this is going to work really well.

Next steps are to

1. look at a larger realistic case
2. return clusters as well as cluster flux assignments
3. distribute the cluster fluxes over the cluster streamlines
4. compare actual solutions and mb errors