1 Balancing Methodology

The primary goal of balancing the SAM is to bring the row and column totals to equivalent values. Additionally, we would also prefer that the entries of the balanced SAM remain similar to their original values in order to keep the values realistic. To do both, we must define an objective function that contains a penalty term for each.

1.1 Residual Penalty

We start by defining a penalty term for the difference in row and column totals. Firstly, our function should penalize negative differences by the same amount as positive differences. Additionally, we want larger differences to be penalized at an increasing rate. Defining our original SAM as $S_{(A)}$ and our balanced SAM as $S_{(B)}$, we may use the L2 norm of the row and column differences:

$$\left\| S_{(B)}\mathbf{1} - S_{(B)}^T\mathbf{1} \right\|_2$$

That is, we subtract the row totals, $S_{(B)}\mathbf{1}$, by the column totals, $S_{(B)}^T\mathbf{1}$, and then sum the squares of each difference. However, we also want to penalize differences independently from the scale of the SAM; so, for instance, if all the entries were halved, this penalty term should not change. Defining μ_A as the average of the entries in the original SAM and μ_B as that of the balanced SAM, we modify this function to:

$$(\mu_A/\mu_B)^2 \cdot \left\| S_{(B)}\mathbf{1} - S_{(B)}^T\mathbf{1} \right\|_2$$

This prevents the optimization algorithm from behaving inappropriately by trying to scale down the SAM to reduce this penalty term. From here on, we refer to this function as the residual penalty.

1.2 Change Penalty

Next, we define a penalty term for the change in the entries. Firstly, we must define change in the SAM using ratios, since the relationships in the SAM are generally defined in percentage terms. For example, given some arbitrary industry, we may want to keep the capital-labor ratio close to their original ratio in the SAM. If we approached this problem by directly taking the sum of squares, we may not keep this ratio constrained appropriately. That is, suppose the original values in the SAM for this particular industry were \$7 for transfers to capital and \$3 for transfers to labor. Changing these values to (\$5, \$3) would produce the same sum of squares error as would the values (\$7, \$5). However, in the first case, the ratio would be 29% lower than the original ratio, while, in the latter case, the ratio would be 40% lower. Hence, we need to approach penalizing change in a different way.

Alternatively, we could compute, for each element, the difference ratio between the new and old values, and then minimize the sum of squares of these terms. To motivate this, let us first define the vector $A \in R^{n^2}$ where n is the number of rows in our SAM; A is a vector of the stacked columns of our original SAM $S_{(A)}$. Likewise, let us refer to B as a vector representation of the balanced SAM. Now, suppose that our unbalanced SAM A is equal to the true balanced SAM B multiplied an normally-distributed error term. That is, the ratio between any arbitrary elements of the two SAMS is equal to:

$$\forall i,\ B_i/A_i - 1 = \varepsilon \qquad \varepsilon \sim N(0, \sigma^2)$$

The maximum likelihood estimator for this equation is equivalent to minimizing the following term:

$$\|B_i/A_i - \mathbf{1}\|_2$$

which is simply the sum of squares of the difference of the ratios from 1. Note that, because this penalty term targets change in the entries, this function puts its minimal penalty on letting $B = A$. Additionally, to accommodate the capital-labor ratio problem mentioned earlier, we are directly minimizing the change in ratios through this function; moreover, this is penalization at an increasing rate. Finally, since each change in ratio is squared, positive and negative changes are penalized equally. In short, this function – the change penalty – achieves all of our goals.

1.3 Constraints

Also, we must include some basic constraints in our model. To start, zero entries must be constrained at zero; additionally, we should exclude zero entries from our change penalty function, since such cases would be undefined in terms of change. Additionally, we must constrain the sign of our entries to their original signs. Positive entries should stay positive and likewise for negative entries. Lastly, we may want some terms to remain the same. For example, we may have a case where knew that certain elements of our original SAM were correct. Hence, we would constrain those elements from changing.

1.4 Objective Function

Finally, we define our objective function as a combination of the change penalty and the residual penalty. By defining these two terms separately, we can construct a weighting scheme for each one in our objective function:

$$\begin{aligned} \min_{S_{(B)}} Z &= w_1 \left((\mu_A/\mu_B)^2 \cdot \left\| S_{(B)}\mathbf{1} - S_{(B)}^T \mathbf{1} \right\|_2 \right) + w_2 \left(\|B_i/A_i - \mathbf{1}\|_2 \right) \\ \text{s.t.} \quad & \operatorname{sgn}(B_i) = \operatorname{sgn}(A_i) \quad \forall\, i \end{aligned}$$

Then, by setting the weights w_1 and w_2, we can implicitly define a tolerance for the amount of acceptable residual penalty. That is, we can run this optimization algorithm on the original SAM and increase w_1 until the residual penalty falls below a desired tolerance. So, overall, this algorithm keeps the balanced SAM close to its original values while producing a feasible SAM for the CGE.