

## 1. REVISED COEFFICIENTS FOR HYPERPLANES

Assume the equations of the hyperplanes are

$$(1) \quad A * x = b$$

Because all the hyperplanes pass one common point,  $A*x - b = 0$ . But due to the numerical issue, the hyperplanes don't pass the common point strictly. Here we set the  $\epsilon$ , which is a very small value to make

$$(2) \quad [A, -b] * \begin{bmatrix} x \\ 1 \end{bmatrix} = \epsilon$$

Then we get the matrix of residual is

$$(3) \quad residual = \frac{\epsilon}{\begin{bmatrix} x \\ 1 \end{bmatrix}}$$

Now we correct the original coefficients to obtain the revised coefficients  $[\tilde{A}, -\tilde{b}]$  by

$$(4) \quad [\tilde{A}, -\tilde{b}] = [A, -b] * residual$$

## 2. THRESHOLD

There are three scenarios of threshold setting problems in implementing GOP. What we need is a criterion for deciding when a value should be treated as zero. It depends on the specific matrix. There must be consistent in these tolerances.

### 2.1. Scenario 1: Setting threshold in removing replicated hyperplanes with $\tau_1$ .

Before comparing if two hyperplanes are the same, we need normalized the coefficients of the hyperplanes and sort them.

2.1.1. *Normalized coefficients.* For normalization, we divide the coefficients by the largest value to let the biggest value of normalized value is 1.

2.2. **Scenario 2: Setting threshold  $\tau_2$  in get number of regions.** We need make lattice to get the number of regions splitted by all the hyperplanes. Each node in lattice represents a matrix and the linear equations must have a solution.

We found when the eigenvalues are very small, it would be hard to set a tolerance to say how small the number can be taken as zero. To find a best way to do this, we tried difference decompositions including singular value(SVD)decomposition, rank-revealing QR (RRQR) decomposition, LU decomposition, and reduced row echelon form (RREF). We found SVD and RRQR are more robust than the other methods, and they both can be used to determine the rank of a matrix. RRQR is more efficient than SVD. RREF and LU are unreliable. Here we use SVD in python to get the rank of the matrix.

To determine if the linear systems have the same solution, we have the theorem like this. We set  $rank[A, b]$  is the rank for matrix  $[A, b]$ .  $[A_{-i}, b_{-i}]$  is a matrix which removed line  $i$  from matrix  $[A, b]$ , and  $[A_{-j}, b_{-j}]$  is a matrix which removed line  $j$  in matrix  $[A, b]$ . If  $rank[A, b] = rank[A_{-i}, b_{-i}]$ , we say linear equations of matrix  $[A, b]$  has the same solution with linear equations of matrix  $[A_{-i}, b_{-i}]$ . Similarly, if  $rank[A, b] = rank[A_{-j}, b_{-j}]$ , we say linear equations of matrix  $[A, b]$  has the same solution with matrix  $[A_{-j}, b_{-j}]$ . Now we can say linear system  $[A_{-i}, b_{-i}]$  has the same solution of matrix  $[A_{-j}, b_{-j}]$ . Then these two nodes of  $[A_{-i}, b_{-i}]$  and  $[A_{-j}, b_{-j}]$  can be merged into one node in lattice.

Here we also considered using condition numbers to set the threshold. In the field of numerical analysis, the condition number of a function with respect to an argument measures the asymptotically worst case of how much the function can change in proportion to small changes in the argument. A problem with low condition number is said to be well-conditioned, while a problem with a high condition number is said to be ill-conditioned. It doesn't work well by using condition number, sometimes the condition number is too large to take two hyperplanes as one but they actually are very close.

**2.3. Scenario 3: Setting threshold  $\tau_3$  in get  $\theta^B$  list.** To get the  $\theta^B$  list of all the regions, we get a point on one hyperplane without on any other hyperplane. Due to the numerical issue, we need set a threshold  $\tau_3$  to say how close the point can be taken as on the hyperplane. This is important because the number of the  $\theta^B$  we got could be changed when  $\tau_3$  is not set properly.

Note that we used the trick of mixed integer program to pick a point which should be far away from the common point  $\bar{x}$ . This constraint is implemented based on a  $L1$  norm for the variable with smallest absolute value of corresponding coefficient.