- Post-processing for QoI.
  - The post-processing code can reduce QoI errors of a reconstructed data produced by any compressors.
  - It is based on constrained optimization with a set of linear equality constraints.
- Supporting features
  - Data type: Double and single precision floating-point data.
  - Dimensions: Each instance should be converted into a vector.
- Theory
  - Let $\boldsymbol{y}$ is a output instance obtained by the post-processing and $\widehat{\boldsymbol{y}}$ is a reconstructed instance by any compressors.
  - We seek that $\boldsymbol{y}$ should satisfy a set of $K$ linear equality constraints of the form $A\boldsymbol{y} = \boldsymbol{b}$, where $A \equiv [\mathrm{V}^1, \dots, \mathrm{V}^K]^T$ is constraint matrix, $\boldsymbol{y}$ is an input vector, and $\boldsymbol{b} \equiv [b_1, \dots, b_K]^T$ is a vector of true (correct) QoI values that are obtained by the original data.
  - Using constrained optimization, the Lagrangian is $\mathcal{L}(\boldsymbol{y}, \boldsymbol{\lambda}) = d(\boldsymbol{y}, \widehat{\boldsymbol{y}}) + \boldsymbol{\lambda}^T(A\boldsymbol{y} - \boldsymbol{b})$, where $d(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ is a distance measure between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$.
  - We use Bregman divergence for the distance measure. By duality theorem, we obtain the dual $q(\lambda) = \sum_{i,j}\left(-\hat{y}_{ij}\exp\left(-\boldsymbol{\lambda}^T A_{ij}\right) + \hat{y}_{ij}\right) - \boldsymbol{\lambda}^T\boldsymbol{b}$.
  - The objective function is the negative of the dual. We minimize the objective function using Newton's method. We update $\boldsymbol{\lambda}$ by $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \alpha H^{-1}\left(\boldsymbol{\lambda}^k\right)g\left(\boldsymbol{\lambda}^k\right)$, where $\boldsymbol{\lambda}^k$ is $\boldsymbol{\lambda}$ of $k^{\text{th}}$ iteration.
- Preparation of constraints.
  - Preparing a set of linear equality constraints is a key step for post-processing.
  - The target constraints must be represented by the form of $A\boldsymbol{y} = \boldsymbol{b}$.
  - Mathematical step might be required for nonlinear constraints.
  - Example – four QoI of XGC.
  - Some QoI of XGC are nonlinear and they can be converted into the linear forms with some assumptions.
    - Density $n = \sum_{i,j} y_{ij}\mathrm{vol}_{ij}$
    - Parallel flow $u_\parallel = \frac{\sum_{i,j} y_{ij}\mathrm{vol}_{ij}v_j^\parallel}{n}$
    - Perpendicular temperature $T_\perp = \frac{m\sum_{i,j} y_{ij}\mathrm{vol}_{ij}\left(v_i^\perp\right)^2}{2n}$
    - Parallel temperature $T_\parallel = \frac{m\sum_{i,j} y_{ij}\mathrm{vol}_{ij}\left(v_j^\parallel - u_\parallel\right)^2}{2n}$
    - The $\mathrm{vol}_{ij}$, $v_j^\parallel$, $v_i^\perp$, and $m$ are obtained from the meta data of XGC.
  - The nonlinearity of the QoI are mainly from the density in denominator. So we assume that the density is the correct (true) value which means it is obtained from the original data. We can treat the density as a constant.
    - Constraint 1: $\sum_{i,j} y_{ij}\mathrm{vol}_{ij} = n^c$
    - Constraint 2: $\sum_{i,j} y_{ij}\mathrm{vol}_{ij}v_j^\parallel = u_\parallel^c n^c$
    - Constraint 3: $\frac{1}{2}m\sum_{i,j} y_{ij}\mathrm{vol}_{ij}\left(v_i^\perp\right)^2 = T_\perp^c n^c$
  - The parallel temperature $T_\parallel$ requires an additional mathematical step.

- $\frac{m \sum_{i,j} y_{ij} \mathrm{vol}_{ij} \left(v_j^{\parallel} - u\right)^2}{2n} = \frac{m}{2} E\left[\left(v^{\parallel} - u_{\parallel}\right)^2\right] = \frac{m}{2}\left[E\left(\left(v^{\parallel}\right)^2\right) - u_{\parallel} E\left(v_j^{\parallel}\right) + (u_{\parallel})^2\right] = \frac{m}{2}\left[E\left(\left(v^{\parallel}\right)^2\right) - (u_{\parallel})^2\right]$
- $E(\cdot)$ is expected value.
- We can eliminate $u_{\parallel}$ in the $T_{\parallel}$ computation.
- Constraint 4: $\frac{1}{2} m \sum_{i,j} y_{ij} \mathrm{vol}_{ij} \left(v_i^{\parallel 0}\right)^2 = n^c \left(T_{\perp}^c + \frac{m}{2}(u_{\parallel})^2\right)$

- Let $\mathrm{V}_{ij}^1 = \mathrm{vol}_{ij}$, $\mathrm{V}_{ij}^2 = \mathrm{vol}_{ij} v_j^{\parallel}$, $\mathrm{V}_{ij}^3 = \frac{m}{2} \mathrm{vol}_{ij} \left(v_i^{\perp}\right)^2$, and $\mathrm{V}_{ij}^4 = \frac{m}{2} \mathrm{vol}_{ij} \left(v_j^{\parallel}\right)^2$. We obtain the constraint matrix $A = [\mathrm{V}^1, \dots, \mathrm{V}^K]^T$ (each of V matrix is converted into a vector) and $\boldsymbol{b} = \left[n_c, u_{\parallel}^c n^c, T_{\perp}^c n^c, n^c \left(T_{\perp}^c + \frac{m}{2}(u_{\parallel})^2\right)\right]^T$.