Discrete Empirical Interpolation Method (DEIM) and Q-DEIM

As introduced in [1], DEIM is a technique for model reduction, with a purpose to determine a subset of relevant points in a model that give the most necessary information about the function. This subset of points can then be used as interpolation points, to reconstruct a function based on the given data points. This is essentially a method for parameterizing equations by a set of parameters θ.
this is an approach to approximate functions of the form $f(x, \theta)$, where

- $\{x_i\}_{i=1,\dots,N}$ is a set of N spatial coordinates, where each $x_i \in \mathbb{R}^d$ $(i = 1, \dots, N)$, for some $d \in \mathbb{N}$.
- $\{\theta_j\}_{j=1,\dots,M}$ is a set of M parameters, where each $\theta_j \in \mathbb{R}^p$ $(j = 1, \dots, M)$, for some $p \in \mathbb{N}$.

For a fixed $j = 1, \dots, M$, let $\underline{f}(\theta_j) = [f(x_1, \theta_j)\ f(x_2, \theta_j)\ \dots\ f(x_N, \theta_j)]^T \in \mathbb{R}^N$ be a vector-valued function. Furthermore, consider two matrices $\boldsymbol{W} \in \mathbb{R}^{N \times k}, \boldsymbol{S} \in \mathbb{R}^{N \times k}$, such that $rank(\boldsymbol{S}^T \boldsymbol{W}) = rank(\boldsymbol{W}^T \boldsymbol{S}) = k$, where k is the number of measurements needed,

- $\boldsymbol{W} \in \mathbb{R}^{N \times k}$ is a matrix with a set of orthonormal columns,
- $\boldsymbol{S} \in \mathbb{R}^{N \times k}$ is a matrix with columns from the $N \times N$ identity matrix.

Define

$$\mathcal{P} \equiv \boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T \tag{1}$$

The goal is to recover (or approximate) the vector-valued function $\underline{f}(\theta_j)$ satisfying

$$\underline{f}(\theta_j) \approx \mathcal{P} \underline{f}(\theta_j) \text{ for } j = 1, \dots, M \tag{2}$$

Specifically, as outlined in Definition 3.1 in [3], consider $\boldsymbol{S} = \boldsymbol{I}(:, \boldsymbol{p}) \in \mathbb{R}^{N \times k}$, where $\boldsymbol{p}$ is a set of k distinct indices $(1 \leq k \leq N)$. The followings are true:

1. $\boldsymbol{S}^T \underline{f}(\theta_j) = [f(x_1, \theta_j)\ \dots\ f(x_k, \theta_j)]^T \in \mathbb{R}^k$.
   In this case, from equation (2), the information of $\underline{f}(\theta_j)$ on the left hand side is given by a subset of k points $\{x_i\}_{i=1,\dots,k}$.
2. $\mathcal{P} \equiv \boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T$ in equation (1) is an oblique objection [3]. Thus, $\mathcal{P}^2 = \mathcal{P}$,
   since $\mathcal{P}^2 = (\boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T)(\boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T) = \boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1}(\boldsymbol{S}^T \boldsymbol{W})(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T$, by associativity.
3. Additionally, from [3], $\mathcal{P} \equiv \boldsymbol{W}(\boldsymbol{S}^T \boldsymbol{W})^{-1} \boldsymbol{S}^T$ in equation (1) is also called an interpolatory projection, which has the following property: For any $\boldsymbol{z} \in \mathbb{R}^N$, $\boldsymbol{z}(\boldsymbol{p}) = (\mathcal{P}\boldsymbol{z})(\boldsymbol{p})$.

Proof.

For any $\boldsymbol{z} \in \mathbb{R}^N$, $\boldsymbol{z}(\boldsymbol{p}) \in \mathbb{R}^k$ is the vector with k rows corresponding to the k indices given, and thus, $\boldsymbol{z}(\boldsymbol{p}) = \boldsymbol{S}^T\boldsymbol{z}$.

Similarly, since $\mathcal{P}\boldsymbol{z} \in \mathbb{R}^N$, $(\mathcal{P}\boldsymbol{z})(\boldsymbol{p}) = \boldsymbol{S}^T\mathcal{P}\boldsymbol{z}$.

Finally, $\boldsymbol{S}^T = \boldsymbol{S}^T\mathcal{P}$, since $\boldsymbol{S}^T\mathcal{P} = (\boldsymbol{S}^T\boldsymbol{W})(\boldsymbol{S}^T\boldsymbol{W})^{-1}\boldsymbol{S}^T$ by associativity.

Therefore, $\boldsymbol{z}(\boldsymbol{p}) = (\mathcal{P}\boldsymbol{z})(\boldsymbol{p})$. □

4. If $\boldsymbol{W} \in \mathbb{R}^{N\times k}$ is partitioned as follow:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \end{bmatrix}, \text{ where } \boldsymbol{W}_1 \in \mathbb{R}^{k\times k} \text{ and } \boldsymbol{W}_2 \in \mathbb{R}^{(n-k)\times k}, \text{ such that } \boldsymbol{S}^T\boldsymbol{W} = \boldsymbol{W}_1.$$

Then, $\mathcal{P} \equiv \boldsymbol{W}(\boldsymbol{S}^T\boldsymbol{W})^{-1}\boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \end{bmatrix} \boldsymbol{W}_1^{-1}\boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{W}_1\boldsymbol{W}_1^{-1} \\ \boldsymbol{W}_2\boldsymbol{W}_1^{-1} \end{bmatrix} \boldsymbol{S}^T = \begin{bmatrix} \boldsymbol{I}_k & \boldsymbol{0}_{k\times(N-k)} \\ \boldsymbol{W}_2\boldsymbol{W}_1^{-1} & \boldsymbol{0}_{(N-k)\times(N-k)} \end{bmatrix}.$

Algorithm to choose $\boldsymbol{W} \in \mathbb{R}^{N\times k}$ and $\boldsymbol{S} \in \mathbb{R}^{N\times k}$:

1. To compute the basis $\boldsymbol{W} \in \mathbb{R}^{N\times k}$:

   First, generate a training set of parameters $\{\theta_j\}_{j=1,\ldots,M}$.

   Consider the snapshot matrix $\mathcal{M} = [\underline{f}(\theta_1) \ \ldots \ \underline{f}(\theta_M)] \in \mathbb{R}^{N\times M}$,

   where $\underline{f}(\theta_j) = [f(x_1,\theta_j) \ f(x_2,\theta_j) \ \ldots \ f(x_N,\theta_j)]^T \in \mathbb{R}^N$ is a vector-valued function, for $j = 1,\ldots,M$.

   Then, compute the left singular vectors corresponding to the k largest singular values.

   In other words, from the SVD of $\mathcal{M}$ such that $\mathcal{M} = \boldsymbol{U\Sigma V}^T$, choose $\boldsymbol{W} = \boldsymbol{U}(:,1:k)$.

   Notice that this method is similar to Principal Component Analysis (PCA).

2. To compute the subset $\boldsymbol{S} \in \mathbb{R}^{N\times k}$:

   As outlined in [2], we can compute the matrix $\boldsymbol{S}$ by using Column Pivoting QR (CPQR), or Strong Rank Revealing QR (SRRQR) for $\boldsymbol{W}^T$.

   Since $\boldsymbol{W}^T \in \mathbb{R}^{k\times N}$, from the result in CPQR, (or SRRQR), $\boldsymbol{W}^T \begin{bmatrix} \boldsymbol{\Pi}_1 & \boldsymbol{\Pi}_2 \end{bmatrix} = \boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \end{bmatrix}$,

   choose $\boldsymbol{S} = \boldsymbol{\Pi}_1$.

References

[1] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[2] Z. Drmač and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016.

[3] D. C. Sorensen and M. Embree. A deim induced cur factorization. *SIAM Journal on Scientific Computing*, 38(3):A1454–A1482, 2016.