

The **RROMP**y rational interpolation method

D. Pradovera, CSQI, EPF Lausanne – davide.pradovera@epfl.ch

Introduction

This document provides an explanation for the numerical method provided by the class **Rational Interpolant**¹ and daughters, e.g. **Rational Interpolant Greedy**², as well as most of the pivoted approximants³.

We restrict the discussion to the single-parameter case, and most of the focus will be dedicated to the impact of the **functionalSolve** parameter, whose allowed values are

- **NORM** (default): see Section 2.1; allows for derivative information, i.e. repeated sample points.
- **DOMINANT**: see Section 2.2; allows for derivative information, i.e. repeated sample points.
- **BARYCENTRIC_NORM**: see Section 3.1; does not allow for a Least Squares (LS) approach.
- **BARYCENTRIC_AVERAGE**: see Section 3.2; does not allow for a Least Squares (LS) approach.

The main reference throughout the present document is [1].

1 Aim of approximation

We seek an approximation of $u : \mathbb{C} \rightarrow V$, with $(V, \langle \cdot, \cdot \rangle_V)$ a complex⁴ Hilbert space (with induced norm $\|\cdot\|_V$), of the form \widehat{p}/\widehat{q} , where $\widehat{p} : \mathbb{C} \rightarrow V$ and $\widehat{q} : \mathbb{C} \rightarrow \mathbb{C}$. For a given denominator \widehat{q} , the numerator \widehat{p} is found by interpolation (possibly, LS or based on radial basis functions) of $\widehat{q}u$. Hence, here we focus on the computation of the denominator \widehat{q} .

Other than the choice of target function u , the parameters which affect the computation of \widehat{q} are:

- **mus** $\subset \mathbb{C}$ ($\{\mu_j\}_{j=1}^S$ below); for all **functionalSolve** values but **NORM** and **DOMINANT**, the S points must be distinct.
- **N** $\in \mathbb{N}$ (N below); for **BARYCENTRIC.***, N must equal $S - 1$.
- **polybasis0** $\in \{ \text{"CHEBYSHEV"}, \text{"LEGENDRE"}, \text{"MONOMIAL"} \}$; only for **NORM** and **DOMINANT**.

To simplify the notation, we set $E = S - 1$. For simplicity, we will consider only the case of S distinct sample points. One can deal with the case of confluent sample points by extending the standard (Lagrange) interpolation steps to Hermite-Lagrange ones.

The main motivation behind the method involves the modified approximation problem

$$u \approx \mathcal{I}^E \left(\left((\mu_j, \widehat{q}(\mu_j)u(\mu_j)) \right)_{j=1}^S \right) / \widehat{q},$$

where $\widehat{q} : \mathbb{C} \rightarrow \mathbb{C}$ is a polynomial of degree $\leq N \leq E$, and $\mathcal{I}^E : (\mathbb{C} \times V)^S \rightarrow \mathbb{P}^E(\mathbb{C}; V)$ is a polynomial interpolation operator, which maps S samples of a function (which lie in V) to a polynomial of degree $\leq E$ with coefficients in V .

More precisely, let

$$\mathbb{P}^M(\mathbb{C}; W) = \left\{ \mu \mapsto \sum_{i=0}^M \alpha_i \mu^i : \alpha_0, \dots, \alpha_M \in W \right\},$$

¹ `./rromp/reduction_methods/standard/rational_interpolant.py`

² `./rromp/reduction_methods/standard/greedy/rational_interpolant_greedy.py`

³ `./rromp/reduction_methods/pivoted/{t,greedy}/rational_interpolant_*.py`

⁴The inner product is linear (resp. conjugate linear) in the first (resp. second) argument: $\langle \alpha v, \beta w \rangle_V = \alpha \bar{\beta} \langle v, w \rangle_V$.

with M either N or E , and W either \mathbb{C} or V . We set

$$\mathcal{I}^E \left(((\mu_j, \psi_j))_{j=1}^S \right) \Big|_{\mu} = \arg \min_{p \in \mathbb{P}^E(\mathbb{C}; V)} \sum_{j=1}^S \|p(\mu_j) - \psi_j\|_V^2.$$

In **RROMP**y, we compute (LS-)interpolants by employing normal equations: given a basis $\{\phi_i\}_{i=0}^E$ of $\mathbb{P}^E(\mathbb{C}; \mathbb{C})$, we expand

$$p(\mu) = \sum_{i=0}^E c_i \phi_i(\mu)$$

and observe that, for optimality, the coefficients $\{c_i\}_{i=0}^E \subset V$ must satisfy

$$\sum_{j=1}^S \sum_{l=0}^E \overline{\phi_l(\mu_j)} \phi_l(\mu_j) c_l = \sum_{j=1}^S \overline{\phi_i(\mu_j)} \psi_j \quad \forall i = 0, \dots, E,$$

i.e., in matrix form⁵,

$$\underbrace{[c_i]_{i=0}^E}_{\in V^S} = \left(\underbrace{\begin{bmatrix} \overline{\phi_i(\mu_j)} \end{bmatrix}_{i=0, j=1}^{E, S}}_{=: \Phi^H \in \mathbb{C}^{S \times S}} \underbrace{\begin{bmatrix} \phi_i(\mu_j) \end{bmatrix}_{j=1, i=0}^{S, E}}_{=: \Phi \in \mathbb{C}^{S \times S}} \right)^{-1} \underbrace{\begin{bmatrix} \overline{\phi_i(\mu_j)} \end{bmatrix}_{i=0, j=1}^{E, S}}_{=: \Phi^H \in \mathbb{C}^{S \times S}} \underbrace{[\psi_j]_{j=1}^S}_{\in V^S}.$$

In practice the polynomial basis $\{\phi_i\}_{i=0}^E$ is determined by the value of **polybasis0**:

- If **polybasis0** = "**CHEBYSHEV**", then $\phi_k(\mu) = \mu^k$ for $k \in \{0, 1\}$ and $\phi_k(\mu) = 2\mu\phi_{k-1}(\mu) - \phi_{k-2}(\mu)$ for $k \geq 2$.
- If **polybasis0** = "**LEGENDRE**", then $\phi_k(\mu) = \mu^k$ for $k \in \{0, 1\}$ and $\phi_k(\mu) = (2 - 1/k)\mu\phi_{k-1}(\mu) - (1 - 1/k)\phi_{k-2}(\mu)$ for $k \geq 2$.
- If **polybasis0** = "**MONOMIAL**", then $\phi_k(\mu) = \mu^k$ for $k \geq 0$.

The actual denominator \hat{q} is found as

$$\hat{q} = \arg \min_{q \in \mathbb{P}^N(\mathbb{C}; \mathbb{C})} \left\| \frac{d^E}{d\mu^E} \mathcal{I}^E \left(((\mu_j, q(\mu_j)u(\mu_j))_{j=1}^S) \right) \right\|_V \quad (1)$$

(*)

where (*) is a normalization condition (which changes depending on **functionalSolve**) to exclude the trivial minimizer $\hat{q} \equiv 0$.

Broadly speaking, the methods described differ in terms of the constraint (*), as well as of the degrees of freedom which are chosen to represent the denominator q .

2 Polynomial coefficients as degrees of freedom

If the polynomial basis $\{\phi_i\}_{i=0}^E$ is hierarchical (as the three ones above), then the E -th derivative of \mathcal{I}^E coincides with the coefficient c_E , and we have

$$\hat{q} = \arg \min_{q \in \mathbb{P}^N(\mathbb{C}; \mathbb{C})} \left\| \underbrace{[0, \dots, 0, 1]}_E (\Phi^H \Phi)^{-1} \Phi^H [q(\mu_j)u(\mu_j)]_{j=1}^S \right\|_V \quad (2)$$

(*)

Using the Kronecker delta ($\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$), the last term $[q(\mu_j)u(\mu_j)]_{j=1}^S \in V^S$ can be factored into

$$\left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S} [q(\mu_j)]_{j=1}^S = \left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S} \tilde{\Phi} [q_i]_{i=0}^N, \quad (3)$$

⁵The superscript H denotes adjunction (conjugate transposition), i.e. $A^H = \overline{A}^T$.

where $\tilde{\Phi}$ is the $S \times (N + 1)$ matrix obtained by extracting the first $N + 1$ columns of Φ . We remark that we have expanded the polynomial q using the basis⁶ $\{\phi_i\}_{i=0}^N$: $q(\mu) = \sum_{i=0}^N q_i \phi_i(\mu)$, with coefficients $\{q_i\}_{i=0}^N \subset \mathbb{C}$.

Combining (2) and (3), it is useful to consider the $(N + 1) \times (N + 1)$ Hermitian matrix with entries $(0 \leq i, i' \leq N)$

$$G_{ii'} = \left\langle \sum_{j'=1}^S \left((\Phi^H \Phi)^{-1} \Phi^H \right)_{Nj'} (\Phi)_{j'i'} u(\mu_{j'}), \sum_{j=1}^S \left((\Phi^H \Phi)^{-1} \Phi^H \right)_{Nj} (\Phi)_{ji} u(\mu_j) \right\rangle_V. \quad (4)$$

If (\star) is quadratic (resp. linear) in $[q_i]_{i=0}^N$, then we can cast the computation of the denominator as a quadratically (resp. linearly) constrained quadratic program involving G .

2.1 Quadratic constraint

We constrain $[\hat{q}_i]_{i=0}^N$ to have unit (Euclidean) norm. The resulting optimization problem can be cast as a minimal (normalized) eigenvector problem for G in (4). More explicitly,

$$[\hat{q}_i]_{i=0}^N = \arg \min_{\substack{\mathbf{q} \in \mathbb{C}^{N+1} \\ \|\mathbf{q}\|_2=1}} \mathbf{q}^H G \mathbf{q}.$$

2.2 Linear constraint

We constrain $\hat{q}_N = 1$, thus forcing q to be monic, with degree exactly N . Given G in (4), the resulting optimization problem can be solved rather easily as:

$$[\hat{q}_i]_{i=0}^N = \frac{G^{-1} \mathbf{e}_{N+1}}{\mathbf{e}_{N+1}^\top G^{-1} \mathbf{e}_{N+1}}, \quad \text{with } \mathbf{e}_{N+1} = [0, \dots, 0, 1]^\top \in \mathbb{C}^{N+1}.$$

3 Barycentric coefficients as degrees of freedom

Here we assume that the sample points are distinct, and that $N = E$, so that, in particular, $\Phi = \tilde{\Phi}$. We can choose for convenience a non-hierarchical basis, dependent on the sample points, for q and \mathcal{I}^E , taking inspiration from barycentric interpolation:

$$\phi_i(\mu) = \prod_{\substack{j=1 \\ j \neq i+1}}^S (\mu - \mu_j). \quad (5)$$

Since all elements of the basis are monic and of degree exactly N , the minimization problem can be cast as

$$\hat{\mathbf{q}} = \arg \min_{\substack{\mathbf{q} \in \mathbb{C}^{N+1} \\ (\star)}} \left\| \underbrace{[1, \dots, 1]}_S (\Phi^H \Phi)^{-1} \Phi^H \left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S} \Phi \mathbf{q} \right\|_V. \quad (6)$$

At the same time, it is easy to see from (5) that the Vandermonde-like matrix Φ is diagonal, so that

$$(\Phi^H \Phi)^{-1} \Phi^H \left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S} \Phi = (\Phi^H \Phi)^{-1} \Phi^H \Phi \left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S} = \left[u(\mu_j) \delta_{jj'} \right]_{j=1, j'=1}^{S, S},$$

and

$$\hat{\mathbf{q}} = \arg \min_{\substack{\mathbf{q} \in \mathbb{C}^{N+1} \\ (\star)}} \left\| \sum_{i=0}^E u(\mu_{i+1}) q_i \right\|_V. \quad (7)$$

Considering (7), it is useful to define the $S \times S$ Hermitian (“snapshot Gramian”) matrix with entries $(0 \leq i, i' \leq N)$

$$G_{ii'} = \langle u(\mu_{i'+1}), u(\mu_{i+1}) \rangle_V. \quad (8)$$

⁶In theory, nothing prevents us from using different bases for \mathcal{I}^E and q , cf. Section 4.

So, once again, if (\star) is quadratic (resp. linear) in $[q_i]_{i=0}^N$, then we can cast the computation of the denominator as a quadratically (resp. linearly) constrained quadratic program involving G .

Before specifying the kind of normalization enforced, it is important to make a remark on numerical stability. The basis in (5) is actually just a (i -dependent) factor away from being the Lagrangian one (for which $\phi_i(\mu_j)$ would equal $\delta_{(i+1)j}$ instead of

$$\delta_{(i+1)j} \prod_{k \neq i+1} (\mu_j - \mu_k),$$

as it does in our case). As such, it is generally a bad idea to numerically evaluate q starting from its expansion coefficients with respect to $\{\phi_i\}_{i=0}^N$. We get around this by exploiting the following trick, whose foundation is in [2, Section 2.3.3]: the roots of $\hat{q} = \sum_{i=0}^N \hat{q}_i \phi_i$ are the N finite eigenvalues λ of the generalized $(N+2) \times (N+2)$ eigenproblem

$$\text{Det} \left(\begin{bmatrix} 0 & \hat{q}_0 & \cdots & \hat{q}_N \\ 1 & \mu_1 & & \\ \vdots & & \ddots & \\ 1 & & & \mu_S \end{bmatrix} - \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \lambda \right) = 0. \quad (9)$$

This computation is numerically more stable than most other manipulations of a polynomial in the basis (5). Once the roots of \hat{q} have been computed, one can either convert it to nodal form

$$\hat{q}(\mu) \propto \prod_{i=1}^N (\mu - \hat{\lambda}_i), \quad (10)$$

or forgo using \hat{q} completely, in favor of a Heaviside-like approximation involving the newly computed roots $\{\hat{\lambda}_i\}_{i=1}^N$ as poles:

$$\frac{\hat{p}(\mu)}{\hat{q}(\mu)} \rightsquigarrow \hat{r}_0 + \sum_{i=1}^N \frac{\hat{r}_i}{\mu - \hat{\lambda}_i}.$$

See the final paragraph in Section 4 for a slightly more detailed motivation of why the Heaviside form of the approximant might be more useful than the standard rational one \hat{p}/\hat{q} in practice.

3.1 Quadratic constraint

We constrain $[\hat{q}_i]_{i=0}^N$ to have unit (Euclidean) norm. The resulting optimization problem can be cast as a minimal (normalized) eigenvector problem for G in (8). More explicitly,

$$[\hat{q}_i]_{i=0}^N = \arg \min_{\substack{\mathbf{q} \in \mathbb{C}^{N+1} \\ \|\mathbf{q}\|_2=1}} \mathbf{q}^H G \mathbf{q}.$$

3.2 Linear constraint

We constrain $\sum_{i=0}^N \hat{q}_i = 1$, so that the polynomial \hat{q} is monic. Given G in (8), the resulting optimization problem can be solved rather easily as:

$$[\hat{q}_i]_{i=0}^N = \frac{G^{-1} \mathbf{1}_S}{\mathbf{1}_S^\top G^{-1} \mathbf{1}_S}, \quad \text{with } \mathbf{1}_S = [1, \dots, 1]^\top \in \mathbb{C}^S.$$

4 Minor observations

- If $N = E$, normal equations are not necessary to compute \mathcal{I}^E , since Φ is square and can be inverted directly. However, in practical applications, it may be useful to decrease the degree E of the interpolant (which, in our presentation, we kept fixed to $S - 1$ for simplicity) to overcome numerical instabilities which may arise in the (pseudo-)inversion of Φ . If this happens, Φ becomes non-square, and normal equations are the only option.

- For **BARYCENTRIC.***, a specific choice of polynomial basis for \mathcal{I}^E was used to diagonalize the functional. Under the assumptions that the sample points are distinct and that $N = E$, one can employ the quasi-Lagrangian basis (5) to expand \mathcal{I}^E in the other approaches as well, thus simplifying significantly the structure of (1):

$$\hat{q} = \underset{\substack{q \in \mathbb{P}^N(\mathbb{C}; \mathbb{C}) \\ (*)}}{\operatorname{arg\,min}} \left\| \sum_{j=1}^S q(\mu_j) u(\mu_j) \prod_{\substack{j'=1 \\ j' \neq j}}^S \frac{1}{\mu_j - \mu_{j'}} \right\|_V.$$

Numerically, this has repercussions on the computation of the term $(\Phi^H \Phi)^{-1} \Phi^H$ in (4).

- In general, **NORM** and **BARYCENTRIC.NORM** can be expected to be more numerically stable than **DOMINANT** and **BARYCENTRIC.AVERAGE**, respectively. This is due to the fact that the normalization is enforced in a more numerically robust fashion.
- If the snapshots are orthonormalized via **POD**⁷, all the V -inner products (resp. norms) are recast as Euclidean inner products (resp. norms) involving the R factor of the generalized (V -orthonormal) QR decomposition of the snapshots.
- If a univariate rational surrogate is built in the scope of multivariate pole-matching-based pivoted approximation⁸, the rational approximant is converted into a Heaviside/nodal representation when different surrogates are combined. As such, the **BARYCENTRIC.*** approach may be preferable to avoid extra computations, as well as additional round-off artifacts.

References

- [1] D. Pradovera, Interpolatory rational model order reduction of parametric problems lacking uniform inf-sup stability, *SIAM J. Numer. Anal.* 58 (2020) 2265–2293. doi:10.1137/19M1269695.
- [2] G. Klein, Applications of Linear Barycentric Rational Interpolation, PhD Thesis no. 1762, Université de Fribourg (2012).

⁷./rrompy/sampling/engines/sampling_engine_pod.py

⁸./rrompy/reduction_methods/pivoted/*